

On a Dynamical Analysis of Reinforcement Learning in Games: Emergence of Occam's Razor

Karl Tuyls * Katja Verbeeck Sam Maes

Computational Modeling Lab, Department of Computer Science,
Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel

Abstract

Modeling learning agents in the context of Multi-agent Systems requires an adequate understanding of their dynamic behaviour. Usually, these agents are modeled similar to the different players in a standard game theoretical model. Unfortunately traditional Game Theory is static and limited in its usefulness.

Evolutionary Game Theory improves on this by providing a dynamics which describes how strategies evolve over time. In this paper, we discuss three learning models whose dynamics are related to the Replicator Dynamics(RD). We show how a classical Reinforcement Learning(RL) technique, i.e. Q-learning relates to the RD. This allows to better understand the learning process and it allows to determine how complex a RL model should be. More precisely, Occam's Razor applies in the framework of games, i.e. the simplest model (Cross) suffices for learning equilibria. An experimental verification in all three models is presented.

1 Introduction

Reinforcement Learning (RL) is already an established and profound theoretical framework for learning in stand-alone systems. Yet, extending RL to multi-agent systems (MAS) does not guarantee the same theoretical grounding. As long as the environment an agent experiences is stationary, and the agent can experiment enough, RL guarantees convergence to the optimal strategy. In a MAS however, the reinforcement an agent receives, may depend on the actions taken by the other agents, present in the system. Obviously in these environments, the convergence results of RL are lost.

Therefore it is important to fully understand the dynamics of reinforcement learning and the effect of exploration in MAS. For this aim we review the Replicator Dynamics (RD) model from Evolutionary Game Theory (EGT). This model

* Author funded by a doctoral grant of the institute for advancement of scientific technological research in Flanders (IWT)

consists of a system of differential equations describing how a population of strategies evolves over time, and plays a central role in biological and economical models.

Several authors have already noticed that the RD can emerge from simple learning models, [1, 8, 9]. In this paper not only the dynamics of the Cross model and Learning Automata will be a topic of interest, but also the dynamics of Q-learning agents. These dynamics open a new perspective in understanding and fine tuning the learning process in games and more general in MAS. It will become clear that in the framework of games the Cross model is sufficient for learning equilibria. In other words, Occam's Razor can be applied.

The outline of the paper is as follows. In section 2 we elaborate on three RL models in order of their complexity. Section 3 describes the replicator dynamics from EGT. We continue with a section on the dynamics of these models. Section 5 describes the experiments. Finally, we end with a discussion.

2 Reinforcement Learning models

In this section we introduce the three reinforcement learning models under consideration in order of complexity. We start with the Cross model and continue with Learning Automata and Q-learning. After considering these three learning models it will become clear that the Cross model is the most simple model of the three presented. More precisely, it is a less complex model than LA and Q-learning in the sense that you don't have to initialise and finetune as many parameters as with LA and Q-learning. Cross doesn't consider a learning rate, a discount factor and temperature as Q-learning does, nor needs a reward and penalty parameters as LA do.

2.1 The Cross Learning model

The cross learning model is a special case of the standard reinforcement learning model [1]. The model considers several agents playing the same normal form game repeatedly in discrete time. At each point in time, each player is characterized by a probability distribution over her strategy set which indicates how likely she is to play any of her strategies. The probabilities change over time in response to experience. At each time step (indexed by n), a player chooses one of its strategies based on the probabilities which are related to each isolated strategy. As a result a player can be represented by a probability vector:

$$p(n) = (p_1(n), \dots, p_r(n))$$

In case of a 2-player game with payoff matrix U , player k gets payoff U_{ij}^k when player 1 chooses strategy i and player 2 chooses strategy j . Players don't observe each others strategies and payoffs. After each stage they update their probability vector, according to,

$$p_i(n+1) = U_{ij} + (1 - U_{ij})p_i(n) \tag{1}$$

$$p_{i'}(n+1) = (1 - U_{ij})P_{i'}(n) \tag{2}$$

Equation (1) expresses how the probability of the selected strategy (i) is updated and equation (2) expresses how all the other strategies $i' \neq i$ are corrected. The probability vector of $Q(n)$ is updated in an analogous manner. This entire system of equations defines a stochastic update process for the players $\{p^k(n)\}$. This process is called the "Cross learning process" in [1]. Börgers and Sarin showed that in an appropriately constructed continuous time limit, this model converges to the asymmetric, continuous time version of the replicator dynamics, see section 3.

2.2 Learning Automata

A LA formalizes a general stochastic system in terms of states, actions, probabilities (state or action) and environment responses [3]. In a variable structure stochastic automaton¹ action probabilities are updated at every stage using a reinforcement scheme. An automaton is defined by a quadruple $\{\alpha, \beta, \mathbf{p}, T\}$ for which α is the action or output set $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ of the automaton, β is a random variable in the interval $[0, 1]$, \mathbf{p} is the action probability vector of the automaton or agent and T denotes an update scheme. The output α of the automaton is actually the input to the environment. The input β of the automaton is the output of the environment, which is modeled through penalty probabilities c_i with $c_i = P[\beta | \alpha_i], i \in \{1 \dots r\}$.

Important examples of update schemes are linear reward-penalty, linear reward-inaction and linear reward- ϵ -penalty. The philosophy of those schemes is essentially to increase the probability of an action when it results in a success and to decrease it when the response is a failure. The general algorithm consist of two update rules, one to update the probability of the selected action and one for all the other actions.

$$p_i(n+1) = p_i(n) + a(1 - \beta(n))(1 - p_i(n)) - b\beta(n)p_i(n) \quad (3)$$

when α_i is the action taken at time n .

$$p_j(n+1) = p_j(n) - a(1 - \beta(n))p_j(n) + b\beta(n)[(r-1)^{-1} - p_j(n)] \quad (4)$$

when $\alpha_j \neq \alpha_i$.

The constants a and b are the reward and penalty parameters respectively. When $a = b$ the algorithm is referred to as linear reward-penalty (L_{R-P}), when $b = 0$ it is referred to as linear reward-inaction (L_{R-I}) and when b is small compared to a it is called linear reward- ϵ -penalty ($L_{R-\epsilon P}$).

If the penalty probabilities c_i of the environment are constant, the probability vector $\mathbf{p}(n+1)$ is completely determined by probability vector $\mathbf{p}(n)$ and hence $\mathbf{p}(n)_{n>0}$ is a discrete-time homogeneous Markov process. Convergence results for the different schemes are obtained under the assumptions of constant penalty probabilities [3].

¹As opposed to fixed structure learning automata, where state transition probabilities are fixed and have to be chosen according to the response of the environment and to perform better than a pure-chance automaton in which every action is chosen with equal probability.

A multi-agent system can be modeled as an automata game. A game $\alpha(t) = (\alpha^1(t) \dots \alpha^n(t))$ of n automata is a set of strategies chosen by the automata at stage t . Correspondingly the outcome is now a vector $\beta(t) = (\beta^1(t) \dots \beta^n(t))$. At every instance all automata update their probability distributions based on the responses of the environment. Each automaton participating in the game operates without information concerning payoff, the number of participants, their strategies or actions.

2.3 Q-learning

Common reinforcement learning methods, which can be found in [7] are structured around estimating value functions. A value of a state or state-action pair, is the total amount of reward an agent can expect to accumulate over the future, starting from that state. One way to find the optimal policy is to find the optimal value function. If a perfect model of the environment as a Markov decision process is known, the optimal value function can be learned with an algorithm called value iteration. Q-learning is an adaptive value iteration method see [7], which bootstraps its estimate for the state-action value $Q_{t+1}(s, a)$ at time $t + 1$ upon its estimate for $Q_t(s', a')$ with s' the state where the learner arrives after taking action a in state s :

$$Q_{t+1}(s, a) \leftarrow (1 - \alpha)Q_t(s, a) + \alpha(r + \gamma \max_{a'} Q_t(s', a')) \quad (5)$$

With α the usual step size parameter, γ a discount factor and r the immediate reinforcement.

The players could therefore use the algorithm of (5) where the state information s is removed. Solutions are formulated in terms of equilibrium situations for the players.

3 The Replicator Dynamics

In Biology, a simple abstraction of an evolutionary process combines two basic elements: a selection mechanism and a mutation mechanism. The mutation provides variety, while selection favors particular varieties over others. Replicator dynamics highlights the role of selection, it describes how systems consisting of different strategies change over time. They are formalized as a system of differential equations. One important assumption of this model is that each replicator represents one (pure) strategy. This strategy is inherited by all the offspring of the replicator.

The general form of a replicator dynamic is the following:

$$\dot{x}_i = [(A\mathbf{x})_i - \mathbf{x} \cdot A\mathbf{x}]x_i \quad (6)$$

In equation (6), x_i represents the density of strategy i in the population, A is the payoff matrix which describes the different payoff values each individual replicator receives when interacting with other replicators in the population. The state of the population (\mathbf{x}) can be described as a probability vector $\mathbf{x} = (x_1, x_2, \dots, x_R)$ which expresses the different densities of all the different types of replicators in the

population. Hence $(A\mathbf{x})_i$ is the payoff which replicator i receives in a population with state \mathbf{x} and $\mathbf{x} \cdot A\mathbf{x}$ describes the average payoff in the population. The growth rate \dot{x}_i/x_i of the population share using strategy i equals the difference between the strategy's current payoff and the average payoff in the population. For further information we refer the reader to [10, 2].

Now, if we assume there exists a relation between the state of the replicator population \mathbf{x} and the probability distribution for a player $p(n)$, two populations will be required, one for each player. Hence, the game is played between the members of two different populations. As a result, we need two systems of differential equations: one for the first player (p) and one for the second player (q). This setup corresponds to a replicator dynamic for asymmetric games. If $A = B^t$, equation (6) would again emerge.

This translates into the following replicator equations for the two populations:

$$\dot{p}_i = [(A\mathbf{q})_i - \mathbf{p} \cdot A\mathbf{q}]p_i \quad (7)$$

$$\dot{q}_i = [(B\mathbf{p})_i - \mathbf{q} \cdot B\mathbf{p}]q_i \quad (8)$$

As can be seen in equation (7) and (8), the growth rate of the types in each population is now determined by the composition of the other population. Note that, when calculating the rate of change using these systems of differential equations, two different payoff matrices (A and B) are used for the two different players.

4 The Dynamics of Learning Models in Games

4.1 The Cross and LA dynamics

The Cross and LA dynamics are considered in the same section because it will become clear that the Cross model is a special case of LA. In [1] it is proven that the RD emerge from the Cross model, and therefore will also emerge from the LA model.

If it is assumed that $b = 0$ and $a = 1$ in equations (3) and (4), the relation between (1) and (2) with (3) and (4) becomes apparent. In this association, when the reward penalty term $a = 1$, the feedback from the environment $(1 - \beta(n))$ equals the game reward U_{ij}^k . Hence the equations become equivalent. As a result the conditions and implications from the relation between the Cross Learning Model and RD also hold for LA games.

4.2 The Q-learning Dynamics

In this section we briefly² describe the relation between Q-learning and the RD. More precisely we present the dynamical system of Q-learning. These equations are derived by constructing a continuous time limit of the Q-learning model, where Q-values are interpreted as Boltzmann probabilities for the action selection. For

²The reader who is interested in the complete derivation, we refer to [9]

reasons of simplicity we consider games between 2 players. The equations for the first player are,

$$\frac{dx_i}{dt} = x_i \alpha \tau((A\mathbf{y})_i - \mathbf{x} \cdot A\mathbf{y}) + x_i \alpha \sum_j x_j \ln\left(\frac{x_j}{x_i}\right) \quad (9)$$

analogously for the second player, we have,

$$\frac{dy_i}{dt} = y_i \alpha \tau((B\mathbf{x})_i - \mathbf{y} \cdot B\mathbf{x}) + y_i \alpha \sum_j y_j \ln\left(\frac{y_j}{y_i}\right) \quad (10)$$

Equations 9 and 10 express the dynamics of both Q-learners in terms of Boltzmann probabilities. Each agent(or player) has a probability vector over his action set , more precisely x_1, \dots, x_n over action set a_1, \dots, a_n for the first player and y_1, \dots, y_m over b_1, \dots, b_m for the second player.

For a complete discussion on this equations we refer to [9]. Comparing (9) or (10) with the RD in (6), we see that the first term of (9) or (10) is exactly the RD and thus takes care of the selection mechanism, see [10]. The mutation mechanism for Q-learning is therefore left in the second term, and can be rewritten as:

$$x_i \alpha \sum_j x_j \ln(x_j) - \ln(x_i) \quad (11)$$

In equation (11) we recognize 2 entropy terms, one over the entire probability distribution x , and one over strategy x_i .

Relating entropy and mutation is not new. It is a well known fact [5, 6] that mutation increases entropy. In [6], it is stated that the concepts are familiar with thermodynamics in the following sense: the selection mechanism is analogous to *energy* and mutation to *entropy*. So generally speaking, mutations tend to increase entropy. Exploration can be considered as the mutation concept, as both concepts take care of providing variety.

5 Experiments

In this paper, analysis restricts itself to 2-player games for reasons of simplicity. We present a commonly used game categorization [4], which is used for the experiments. general reward tables are,

$$\begin{pmatrix} a_{11}, b_{11} & a_{12}, b_{12} \\ a_{21}, b_{21} & a_{22}, b_{22} \end{pmatrix}$$

Table 1: The a_{ij} defines the payoff for the first player, the b_{ij} defines the payoff for the second player

Subclass 1: if $(a_{11} - a_{21})(a_{12} - a_{22}) > 0$ or $(b_{11} - b_{12})(b_{21} - b_{22}) > 0$, at least one of the 2 players has a dominant strategy, therefore there is just 1 strict equilibrium.

Subclass 2: if $(a_{11} - a_{21})(a_{12} - a_{22}) < 0, (b_{11} - b_{12})(b_{21} - b_{22}) < 0,$ and $(a_{11} - a_{21})(b_{11} - b_{12}) > 0,$ there are 2 pure equilibria and 1 mixed equilibrium.

Subclass 3: if $(a_{11} - a_{21})(a_{12} - a_{22}) < 0, (b_{11} - b_{12})(b_{21} - b_{22}) < 0,$ and $(a_{11} - a_{21})(b_{11} - b_{12}) < 0,$ there is just 1 mixed equilibrium.

The first subclass includes those type of games where each player has a dominant strategy, as for instance the prisoners dilemma. However it includes a larger collection of games since only 1 of the players needs to have a dominant strategy. In the second subclass none of the players has a dominated strategy. But both players receive the highest payoff by both playing their first or second strategy. This is expressed in the condition $(a_{11} - a_{21})(b_{11} - b_{12}) > 0.$ The third subclass only differs from the second in the fact that the players don't receive their highest payoff by both playing the first or the second strategy. This is expressed by the condition $(a_{11} - a_{21})(b_{11} - b_{12}) < 0.$ In the following three subsections we describe the results of the experiments conducted in each subclass.

5.1 General Settings

As a representative for the three subclasses we chose the following games,

$$\begin{pmatrix} 1, 1 & 5, 0 \\ 0.5 & 3, 3 \end{pmatrix}, \quad \begin{pmatrix} 2, 1 & 0, 0 \\ 0, 0 & 1, 2 \end{pmatrix}, \quad \begin{pmatrix} 2, 3 & 3, 1 \\ 4, 2 & 1, 4 \end{pmatrix}$$

Figure 1: the first table represents the Prisoners Dilemma, the second table the battle of the sexes game and the third is randomly chosen according to the conditions of subclass 3

5.2 The Cross and LA experiments

In figure 2 the replicator dynamic of the prisoners dilemma game is plotted using the differential equations of 7 and 8.

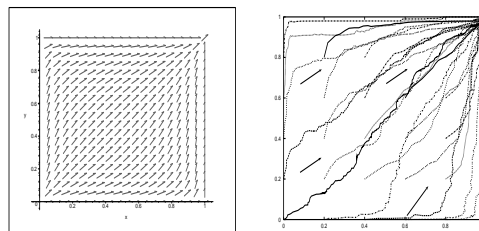


Figure 2: *Left:* The direction field of the RD of the prisoners game. *Right:* The paths induced by the learning process.

More specifically, the figure on the left illustrates the direction field of the replicator dynamic and the figure on the right shows the learning process of LA.

We plotted for both players the probability of choosing their first strategy (in this case defect). As starting points for the LA we chose a grid of 25 points. In every point a learning path starts and converges to the equilibrium at the point $(1, 1)$. As you can see all the sample paths of the reinforcement learning process approximate the paths of the RD.

For games of the second subclass (battle of the sexes) we have

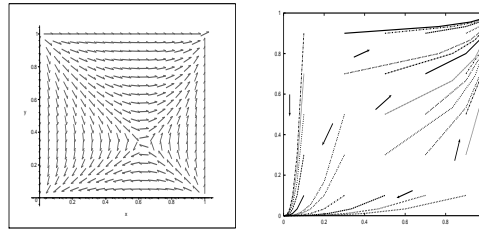


Figure 3: *Left*: The direction field of the RD of the battle of the sexes game. *Right*: The paths induced by the learning process.

Here you can see two pure equilibria at $(0, 0)$ and at $(1, 1)$, and one mixed at $(2/3, 1/3)$. Now we have convergence to the 2 strict equilibria. The third equilibrium is very unstable as you can see in the direction field plot. This instability is the reason why it will not emerge from the learning process on the long run. For games of the third subclass 3 we have,

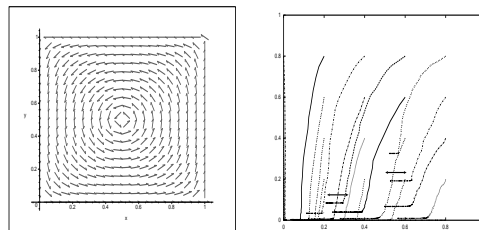


Figure 4: *Left*: The direction field of the RD. *Right*: The paths induced by the learning process.

As can be seen the behavior is not completely the same for this type of game as that of the replicator dynamics. It is a known fact that the asymptotic behaviour of the learning process can differ from that of the replicator dynamics [1].

5.3 The Q-learning experiments

We only describe the experiments of subclass 2. All the experiments can be found in [9]. Nevertheless for all categories the learning dynamics will converge to the system of differential equations. The important aspect is that obtaining convergence to a Nash equilibrium with Q-learning is more cumbersome than with Cross.

In figure 5 the direction field plot of the differential equations of this game is plotted. Again the direction field of the equations are plotted for 3 values of τ , more precisely 10, 100, 1000. In the first 2 plots τ isn't big enough to reach for one of the three Nash equilibria. Only in the last one the dynamics attain the Nash equilibria (the 3 attractors in the last plot) for the game at the coordinates $(1, 1)$, $(0, 0)$ and $(\frac{2}{3}, \frac{1}{3})$. The mixed equilibrium though is very unstable. Any small perturbation away from this equilibrium will typically lead the dynamics to one of the 2 pure equilibria.

In figure 6 we also plotted the Q-learning process for the same game with the same settings as for the system of differential equations. In the chosen points a learning path starts and converges to a particular point. If you compare the plots with the direction field plots for the same value of τ you can see that the sample paths of the learning process approximates the paths of the differential equations. The instability of the mixed equilibrium is the reason why this equilibrium doesn't emerge from the learning process.

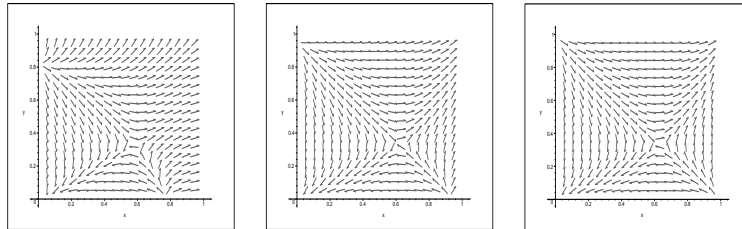


Figure 5: The direction field plots of the battle of the sexes (subclass 2) game with $\tau = 10, 100, 1000$

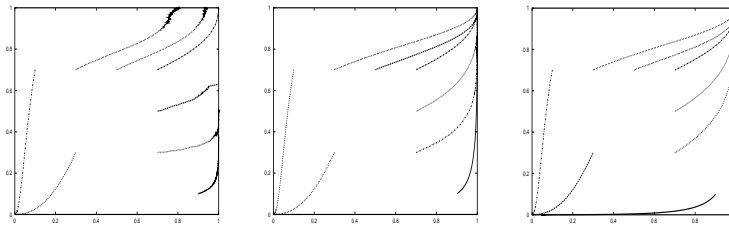


Figure 6: The Q-learning plots of the battle of the sexes (subclass 2) game with $\tau = 10, 100, 1000$

6 Discussion

The major contribution of this paper is that with a theoretical analysis of the dynamics of different reinforcement learning algorithms can be shown that in the

context of games the Cross model suffices to attain the right equilibria. In other words, Occam's Razor can be applied to the field of reinforcement learning in games. It turns out that the Cross model keeps things most simple in the sense of setting parameters and computational effort. The experiments confirm that with the Cross model, the Nash equilibria can be reached in the most elegant way. As opposed to LA, where a lot of tuning is needed with setting the reward and penalty variables to the correct values. Also Q-learning demands a lot of finetuning as can be seen in the experiments 6. There is the temperature τ , the learning rate α and the discount factor γ . In the context of learning agents in games, there is no need to complicate the learning algorithm more than the Cross learning model. Another interesting remark is that learning can be very time consuming, especially when you need to fine tune some parameters. As the experiments illustrate, plotting the direction field of the dynamical system of the learning model beforehand gives information on how to initialise the learning agents so that they end up in the most interesting attractors of the game.

References

- [1] Börgers, T., Sarin, R., Learning Through Reinforcement and Replicator Dynamics. *Journal of Economic Theory*, Volume 77, Number 1, November 1997.
- [2] Hofbauer, J., Sigmund, K., *Evolutionary Games and Population Dynamics*, Cambridge University Press, 1998.
- [3] Narendra, K., Thathachar, M., *Learning Automata: An Introduction*. Prentice-Hall (1989).
- [4] Redondo, F.V., *Game Theory and Economics*, Cambridge University Press, (2001).
- [5] Schneider, T.D., Evolution of biological information. *journal of NAR*, volume 28, pages 2794 - 2799, 2000.
- [6] Stauffer, D., *Life, Love and Death: Models of Biological Reproduction and Aging*. Institute for Theoretical physics, Köln, Euroland, 1999.
- [7] Sutton, R.S., Barto, A.G. : *Reinforcement Learning: An introduction*. Cambridge, MA: MIT Press (1998).
- [8] Tuyls, K., Lenaerts, T., Verbeeck, K., Maes, S. and Manderick, B, Towards a Relation Between Learning Agents and Evolutionary Dynamics. *Proceedings of BNAIC 2002*. KU Leuven, Belgium.
- [9] Tuyls, K., Verbeeck, K. and Lenaerts, T., A Selection-Mutation model for Q-learning in MAS. Accepted at AAMAS 2003. Melbourne, Australia.
- [10] Weibull, J.W., *Evolutionary Game Theory*, MIT Press, (1996).