Dynamic Analysis of Multiagent Q-learning with ϵ-greedy Exploration

Eduardo Rodrigues Gomes EGOMES@GROUPWISE.SWIN.EDU.AU
Ryszard Kowalczyk RKOWALCZYK@GROUPWISE.SWIN.EDU.AU
Swinburne University of Technology, John Street, Hawthorn, VIC 3122, Australia

Abstract

The development of mechanisms to understand and model the expected behaviour of multiagent learners is becoming increasingly important as the area rapidly find application in a variety of domains. In this paper we present a framework to model the behaviour of Q-learning agents using the ϵ-greedy exploration mechanism. For this, we analyse a continuous-time version of the Q-learning update rule and study how the presence of other agents and the ϵ-greedy mechanism affect it. We then model the problem as a system of difference equations which is used to theoretically analyse the expected behaviour of the agents. The applicability of the framework is tested through experiments in typical games selected from the literature.

1. Introduction

Multiagent Learning (MAL) has become one of the most active areas of Artificial Intelligence. As MAL-based systems find application in wide variety of domains, the development of mechanisms to understand and model the expected behaviour of multiagent learners is becoming increasingly important. The advantages of having such mechanisms are many. For instance, MAL systems usually have parameters that need to be adjusted so the overall behaviour of the system can be optimized. The usual approach to setup those parameters is to execute extensive experimentation with different configurations and to aggregate the outcomes in the hope of finding some useful information (Vidal & Durfee, 2003). A better understanding of the expected behaviour can help the system's designer in the task.

In this paper we investigate the case of Multiagent Q-learning with ϵ-greedy exploration. Q-learning is certainly one of the most studied Reinforcement Learning (RL) algorithm and has been applied with success in several domains, from relatively simple toy problems, such as Cliff-Walking (Sutton & Barto, 1998), to more complex ones, such as web-based education (Iglesias et al., 2008) and face recognition (Harandi et al., 2008). Initially proposed for single-agent environments, the simplicity and effectiveness of this algorithm has led to its application also in multiagent configurations, for example Galstyan et al. (2004) and Ziogos et al. (2007). In this case, however, its supporting theoretical framework and convergence guarantees are lost.

One of the difficulties of the multiagent case is to cope with the very dynamic environment generated by multiple learners. There is also the co-adaptation effect, in which one agent adapts its strategy to the others', and vice-versa, in a cyclic fashion. In addition, the rewards that one agent receives depend on the actions of the other agents. All these features make it especially difficult to predict and to model the learning behaviour (Panait & Luke, 2005).

An important research in the area is the work of Tuyls et al. (2003). The authors studied the case of Q-learning agents with *Boltzmann* exploration. They developed a continuous time model for the learning process and have shown a link between the model and the Replicator Dynamics (RD) of Evolutionary Game Theory (Hofbauer & Sigmund, 1998). The main principle of the RD is that the growth in the probability of playing a given action is directly proportional to the performance of that action against the others. The ϵ-greedy mechanism, however, produces different dynamics. This mechanism defines a semi-uniform probability distribution in which the current best action is selected with probability $1 - \epsilon$ and a random action

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

with probability ϵ. Hence, that research cannot be directly applied in our case.

The importance of obtaining a model for Multiagent Q-learning with ϵ-greedy exploration is justified through its large number of applications. For example: Galstyan et al. (2004) applies the algorithm to develop a decentralized resource allocation mechanism; Gomes and Kowalczyk (2007) study the problem of learning demand functions; and Ziogos et al. (2007) investigate the development of bidding strategies.

Therefore, in this paper we present a framework to model the dynamics of Multiagent Q-learning with the ϵ-greedy exploration mechanism. For this, we analyse a continuous-time version of the Q-learning update rule and study how the ϵ-greedy mechanism and the presence of other agents affect it. We then use this analysis to model the problem as a system of difference equations which is used to calculate the expected evolution of the Q-values and, consequently, the expected behaviour of the agents.

The paper is organized as follows. In the next section we review the Q-learning algorithm with ϵ-greedy exploration and its extension to multiagent scenarios. In Section 3 we present the analysis and the equations to model the behaviour of the agents. In Section 4 we provide the evaluation of the framework. We compare the theoretical behaviour obtained by the model with the behaviour found in real experimentation with Q-learning. Section 5 discusses some related works and Section 6 concludes the paper.

2. Background

In this section we briefly review the Q-learning algorithm, the ϵ-greedy action-selection mechanism and the extension of Q-learning to multiagent scenarios.

2.1. Single-agent Q-learning

The task of a Q-learning agent is to learn a mapping from environment states to actions so as to maximize a numerical reward signal (Sutton & Barto, 1998). The model is formalized by a tuple (S, A, T, R), where S is a discrete set of environment states, A is a discrete set of actions, T is a state transition function $S \times A \times S \rightarrow [0, 1]$, and R is a reward function $S \times A \rightarrow \mathbb{R}$. One of the attractives of Q-learning is that it assumes no knowledge about state transitions and reward functions, which must be learned from the environment. In each step, the agent receives a signal from the environment indicating its state $s \in S$ and chooses an action $a \in A$. Once the action is performed, it changes the state of the environment, generating a reinforcement signal $r \in R$ that is then used to evaluate the quality of the decision by updating the corresponding $Q(s, a)$ values.

The $Q(s, a)$-values are estimations of the $Q^*(s, a)$-values, which represent the sum of the immediate reward obtained by taking action a at state s and the total discounted expected future rewards obtained by following the optimal policy thereafter. By updating $Q(s, a)$, the agent eventually makes it converge to the $Q^*(s, a)$. The optimal policy is then followed by selecting the actions where the Q^*-values are maximum. The formula used to update the Q-values is:

$$Q(s,a) := Q(s,a) + \alpha(r(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

where $0 < \alpha < 1$ is the learning rate and $0 < \gamma < 1$ is the discount rate.

Considering that the probabilities of making state transitions T and receiving specific reinforcement signals R do not change over time, i.e. a stationary environment, if each action is executed in each state an infinite number of times and α is decayed appropriately, the Q-values will converge with probability 1 to the optimal ones (Sutton & Barto, 1998).

2.2. ϵ-greedy Mechanism

An important component of Q-learning is the action selection mechanism. This mechanism is responsible for selecting the actions that the agent will perform during the learning process. Its purpose is to harmonize the trade-off between exploitation and exploration such that the agent can reinforce the evaluation of the actions it already knows to be good but also explore new actions.

In this paper we consider the ϵ-greedy exploration. This mechanism selects a random action with probability ϵ and the best action, i.e. the one that has the highest Q-value at the moment, with probability 1-ϵ. As such, it can be seen as defining a probability vector over the action set of the agent for each state. If we let $\mathbf{x} = (x_1, x_2, ..., x_j)$ be one of these vectors, then the probability x_i of playing action i is given by:

$$x_i = \begin{cases} (1-\epsilon) + (\epsilon/n), & \text{if } Q \text{ of } i \text{ is the highest} \\ \epsilon/n, & \text{otherwise} \end{cases}$$

where n is the number of actions in the set.

2.3. Multiagent Q-learning

Multiagent Q-learning is a natural extension of single-agent Q-learning to multiagent scenarios. In this ap-

proach, the agents are equipped with a standard Q-learning algorithm each and learn independently without considering the presence of each other in the environment. The rewards and the state transitions, however, depend on the joint actions of all agents. The problem is formalized as a tuple $(n, S, A_{1\cdots n}, T, R_{1\cdots n})$, where n is the number of players, S is the set of states, A_i is the set of actions available to agent i with A being the joint action space $A_1 \times \cdots \times A_n$, T is the transition function $S \times A \times S \rightarrow [0, 1]$, and R_i is the reward function for the ith player $S \times A \rightarrow \mathbb{R}$. Note that both T and R are defined over the joint action space.

3. A Model of Multiagent Q-learning

We now present our model for multiagent Q-learning with ϵ-greedy exploration. To develop this model, in Section 3.1 we study how the ϵ-greedy mechanism and the presence of other agents affect the learning process of one agent. For this, we first show the derivation of a continuous time equation for the Q-learning rule. We then analyse the limits of this equation for the case of a single learner and show how they change dynamically when multiple learners are considered. Finally, we show how the ϵ-greedy mechanism affects the shape of the modelled function. The observations and results from this study are used in Section 3.2 to develop a system of difference equations to model the behaviour of the learners.

For simplicity of explanation, we consider scenarios composed of 2 agents with 2 actions each and a single state. The reward functions of the agents in this case can be described using payoff tables of the form:

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

where A describes the rewards, or payoffs, for the first agent and B the rewards for the second. Given the existence of only one state, the Q-learning update rule can be simplified to

$$Q_{a_i} := Q_{a_i} + \alpha(r_{a_i} - Q_{a_i}) \tag{1}$$

where Q_{a_i} is the Q-value of agent a for action i and r_{a_i} is reward that agent a receives for executing action i. Please note that this notation is slightly different from the notation applied in Section 2.

3.1. Analysis

We start the study by rewriting the update rule for the first agent as follows:

$$Q_{a_i}(k+1) - Q_{a_i}(k) = \alpha(r_{a_i}(k+1) - Q_{a_i}(k)) \tag{2}$$

This difference equation describes the absolute growth in Q_{a_i} between times k and $k+1$. To obtain its continuous time version, consider $\Delta t \in [0, 1]$ to be a small amount of time and $Q_{a_i}(k + \Delta t) - Q_{a_i}(k) \approx \Delta t \times \alpha(r_{a_i}(k+\Delta t) - Q_{a_i}(k))$ to be the approximate growth in Q_{a_i} during Δt. Note that this equation becomes: an identity when $\Delta t = 0$; Equation 2 when $\Delta t = 1$; and a linear approximation when Δt is between 0 and 1. Dividing both sides of the equation by Δt, $\frac{Q_{a_i}(k+\Delta t) - Q_{a_i}(k)}{\Delta t} \approx \alpha(r_{a_i}(k + \Delta t) - Q_{a_i}(k))$, and taking the limit for $\Delta t \rightarrow 0$, $\lim_{\Delta t \rightarrow 0} \frac{Q_{a_i}(k+\Delta t) - Q_{a_i}(k)}{\Delta t} \approx \alpha(r_{a_i}(k) - Q_{a_i}(k))$, we obtain

$$\frac{dQ_{a_i}(k)}{dt} \approx \alpha(r_{a_i}(k) - Q_{a_i}(k)) \tag{3}$$

which is an approximation for the continuous time version of Equation 2. This result is in line with Tuyls et al. (2003).

The general solution for Equation 3 can be found by integration:

$$Q_{a_i}(k) = Ce^{-\alpha t} + r_{a_i} \tag{4}$$

where C is the constant of integration. As e^{-x} is a monotonic function and $\lim_{x \rightarrow \infty} e^{-x} = 0$, it is easy to observe that the limit of Equation 4 when $t \rightarrow \infty$ is r_{a_i}:

$$\lim_{t \rightarrow \infty} Q_{a_i}(k) = \underbrace{\lim_{t \rightarrow \infty} Ce^{-\alpha t}}_{0} + \underbrace{\lim_{t \rightarrow \infty} r_{a_i}}_{r_{a_i}} = r_{a_i}$$

If we consider that only the first agent is learning and that the second is using a pure strategy, and assuming that the rewards are noise-free, playing a particular action will always generate the same reward for the first agent. In this case, the derivation above is enough to confirm that Q_{a_i} will monotonically increase or decrease towards r_{a_i}, for any initial value of Q_{a_i}. More specifically, the function is monotonically increasing if $Q_{a_i}(0) < r_{a_i}$ and monotonically decreasing if $Q_{a_i}(0) > r_{a_i}$.

If the second agent is using a mixed strategy and the game is played repeatedly, then r_{a_i} can be replaced by

$$E[r_{a_i}] = \sum_j a_{ij} y_j \tag{5}$$

which is the expected payoff of the first agent given the mixed strategy $\mathbf{y}$ of the second. Note that a pure strategy is the specific case of a mixed strategy in which probability 1 is given to one of the actions. We then rewrite Equation 3 and 4 respectively as

$$\frac{dQ_{a_i}(k)}{dt} \approx \alpha(E[r_{a_i}(k)] - Q_{a_i}(k)) \tag{6}$$

$$Q_{a_i}(k) = Ce^{-\alpha t} + E[r_{a_i}] \tag{7}$$

Thus, if the adversary is not learning, Q_{a_i} will move in expectation towards $E[r_{a_i}]$ in a monotonic fashion. With a learning adversary, however, the situation is more complex. In this case, there is a possibility that the expected reward will change over time. A learning adversary can change its probability vector, which affects the expected reward. If we first look at Equation 6, changes in the expected reward will modify the associated direction field and, consequently, the equilibrium points of it. At this level, every time the expected reward changes, a new direction field is generated. If we now look at Equation 7, the changes will modify the limit and, possibly, the direction of Q_{a_i}. Hence, it is important to identify when they will occur.

The ϵ-greedy mechanism updates the probability vector whenever a new action becomes the one with the highest Q-value. Thus, we need to identify the intersection points in the functions of the adversary. It follows that the overall behaviour of the agent depends on these intersection points as they define which values Q_{a_i} will converge to.

From the analysis point of view, the fact that the expected rewards can change over time implies that Equation 6 cannot be solved in the same way we solved Equation 3. However, one can easily derive the paths given the initial Q-values.

Another important aspect to be considered in the model is the *speed* in which the Q-values are updated. During the learning process, the actions have different probabilities of being played. For example, if $\epsilon = 0.2$, the Q-value of the current best action has a probability of 0.9 of being updated, while the other has a probability of 0.1 (considering a 2-actions game). It means that the Q-values are updated at different *speeds*. To simulate this behaviour, we define the growth in the Q-values as directly proportional to the probabilities. Then, Equation 6 becomes

$$\frac{dQ_{a_i}(k)}{dt} \approx x_i(k)\alpha(E[r_{a_i}(k)] - Q_{a_i}(k)) \tag{8}$$

where $x_i(k)$ is the probability of playing action i at time k.

It is important to emphasize that the *speed* of the updates affects the shape of the functions and, as a consequence, the points at which they will intersect each other. As such, this component plays a very significant role in the model. Roughly speaking, the expected reward indicates the values Q_{a_i} will converge to, the *speed* of the updates defines the paths that it will follow to get there and the presence of intersection points in the functions of the adversary determines if it is ever going to get there.

It should be clarified, however, that while the presence of intersection points in one agent's function does not affect the limits of its equations and the equilibrium points of the associated slope fields, it does affect the speed of the convergence and the slope field itself. To illustrate it, suppose that x_i and $E[r_{a_i}]$ are constants. Then, by integration we can find the general solution for Equation 8:

$$Q_{a_i}(k) = Ce^{-x_i\alpha t} + E[r_{a_i}] \tag{9}$$

Note that the only difference between this equation and Equation 7 is the exponential term. Because the limit of this term is 0 for $t \to \infty$, the limit of the equation remains $E[r_{a_i}]$, regardless of the value of x_i. On the other hand, different values of x_i generate different slope fields. This can be seen in Figure 1 where we plotted the slope fields obtained when $E[r_{a_i}] = 5$ and $\alpha = 0.2$ for $x_i \in \{0.1, 0.9\}$. For the sake of comparison, we have also plotted the sample paths for $Q_{a_i}(0)$ equal to 0, 2, 8 and 10.

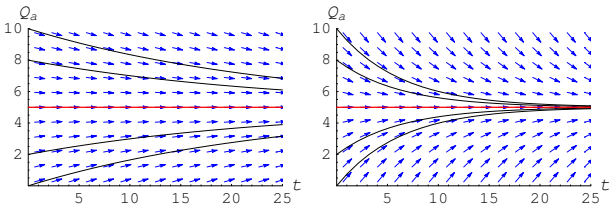

Figure 1. Slope fields associated with Equation 8 ($\alpha = 0.2$ and $r_{a_i} = 5$) for $x_i = 0.1$ (Left) and $x_i = 0.9$ (Right), and examples of specific solutions obtained when $Q_{a_i}(0) \in \{0, 2, 8, 10\}$.

3.2. The model

For the first and the second player, respectively, let A and B be the payoff matrices, $\mathbf{x}$ and $\mathbf{y}$ be the probability vectors, and Q_a and Q_b be the vectors of Q-values. Then, based on the analysis presented in the previous section, the expected behaviour for the Q-values can be modelled by the system of equations:

$$
\begin{aligned}
Q_{a_i}(k+1) &= Q_{a_i}(k) + x_i(k)\alpha(\sum_j a_{ij}y_j(k) - Q_{a_i}(k)) \\
Q_{b_i}(k+1) &= Q_{b_i}(k) + y_i(k)\alpha(\sum_j b_{ij}x_j(k) - Q_{b_i}(k)) \\
x_i(k) &= \begin{cases} (1-\epsilon) + (\epsilon/n), & \text{if } Q_{a_i}(k) \text{ is the highest} \\ \epsilon/n, & \text{otherwise} \end{cases} \\
y_i(k) &= \begin{cases} (1-\epsilon) + (\epsilon/n), & \text{if } Q_{b_i}(k) \text{ is the highest} \\ \epsilon/n, & \text{otherwise} \end{cases}
\end{aligned} \tag{10}
$$

Having the above model for the Q-values, the expected behaviour of the agents can be derived by tracking the actions with highest Q-value over the learning process of each agent. In the next sections the applicability of the framework is tested through experiments in typical games from the literature.

4. Application of the Model in Typical Games

The present section illustrates the application of the framework in two games selected from the literature: the Prisoners Dilemma and an interesting game from Tuyls et al. (2003). For each game we compare the theoretical behaviour obtained with the model with the behaviour found in real experimentation with Q-learning. The experiments were performed with the same configuration as for the model and the results aggregated with the statistical *median*. The *median* is employed because it is more robust in the presence of outlier values than the mean. Therefore, it is more informative in showing the typical Q-values found during the experiments.

4.1. The Prisoners Dilemma

The first game we consider is the Prisoners Dilemma (PD). This game has a single Nash Equilibrium in which both players play their dominant strategies (action 1). The payoff matrices for the first and second players are respectively

$$
\mathrm{A} = \begin{bmatrix} 1 & 5 \\ 0 & 3 \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} 1 & 0 \\ 5 & 3 \end{bmatrix}
$$

In Figure 2 we plot the graphs obtained for this game when the initial Q-values are set to $Q_a = [0, 1]$ and $Q_b = [1, 0]$, and the learning parameters set to $\alpha = 0.1$ and $\epsilon = 0.4$. The starting strategies of the agents given these configurations are $\mathbf{x} = [0.2, 0.8]$ and $\mathbf{y} = [0.8, 0.2]$. The graphs on the left-hand side of the figure compare the theoretical curves of Q-values obtained by the model with the curves found in the experiments. The experimental curves show the median Q-values over 5000 learning experiments. The graphs in the center and on the right-hand side show respectively the theoretical and the observed dynamics for the strategies of the agents. The first is obtained from the analysis of the theoretical Q-values and the second from the analysis of the median Q-values.

We first analyse the learning dynamics from the perspective of agent 2. The Q-values of this agent in the beginning of the learning process describe curves that would converge to 4.2 and 2.4 if no intersection point had been found in the curves of agent 1. The values are the expected rewards of agent 2 given the strategy of agent 1 in that period. It can be seen that the curve of action 1 is quicker than the curve of action 2. This behaviour is linked to the starting strategy of this agent, which allocates probability 0.8 for the first action and 0.2 for the second. Just after time 20, there is a change in the direction of the curves. This change results from a change in the expected rewards, generated by the new strategy adopted by agent 1 after the intersection point found in its curves. From that point on, the curves of agent 2 start to converge towards 1.8 and 0.6, the new expected rewards, and eventually stabilize around these values. Meanwhile, the Q-values of agent 1 evolve constantly towards 1.8 and 0.6. The intersection point does not affect its expected rewards but changes the *speed* of the convergence and consequently the shape of its curves.

As seen in the graphs, the model is able to capture all the major trends found in the experiments. One particular point to note, however, is that while the changes in the theoretical curves are very crisp, in the observed ones they are actually smoother. The explanation for this behaviour is that the intersections in the experiments do not take place all the time in the exact point found by the model. The main aspect affecting the location of the intersection point is the *speed* of the updates, which is in fact a result of the stochastic process. It follows that, in our example, the strategy of agent 1 can change before or after the theoretical point, smoothing the curves when the median is calculated.

Figure 3 shows an example of the typical behaviour found in the experiments. Note that the intersection in the curves of the agent 1 was found slightly after the theoretical point. Also note that, apart from the local variabilities generated by the stochasticity in the actions of both agents, the general trends of the curves match the trend found by the model very well.

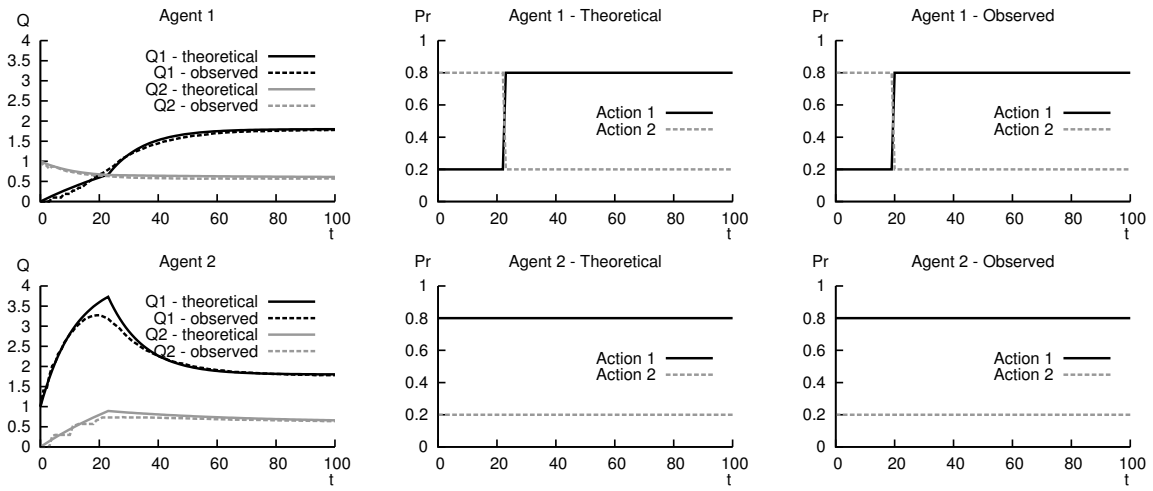

Figure 2. Prisoner's Dilemma: comparison between the theoretical Q-values derivated by the model and the median Q-values observed in the experiments (left); the expected dynamics for the agents' strategies according to the model (center); and the dynamics observed in the experiments (right).

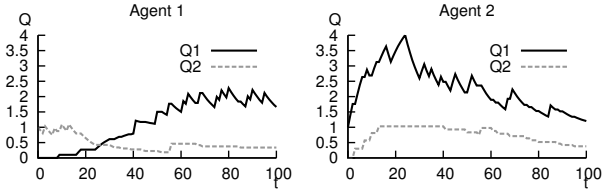

Figure 3. Example of an individual run of Q-learning in the Prisoners Dilemma.

4.2. A Game with no Pure Equilibrium

The second game we consider has been selected from Tuyls et al. (2003). It has no Nash Equilibrium in pure strategies and a unique Nash Equilibrium in mixed strategies, where both players play the first action with probability 0.5. The payoff tables are as follows:

$$\mathrm{A} = \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix} \qquad \mathrm{B} = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}$$

In Figures 4 we plot the graphs obtained when the initial Q-values are $Q_a = [0, 1]$ for agent 1 and $Q_b = [2, 3]$ for agent 2. The learning parameters are $\alpha = 0.1$ and $\epsilon = 0.1$. The experimental results show the median Q-values over 5000 learning experiments.

The results for this example can be divided in two parts. The first part is characterized by major changes in the curves of both agents, which the model is able to capture very well. In particular, note that the Q-values of agent 2 stabilize in the very beginning of the learning process. Around time 100, however, there is an intersection point in the curves of agent 1 that violates the equilibrium and triggers the process of adaptations.

In the second part there seems to be a discrepancy between the model and the experiments. According to the model, this part is characterized by a cycle-like behaviour, indicating that the strategies will not stabilize. Instead, the experimental results show the convergence of the system.

To further illustrate this case, Figure 5 shows an example of an individual run of the Q-learning algorithm for this scenario. The graphs reveal that the experiments actually present a cyclic behaviour similar to the one described by the model. The convergence shown in Figure 4 is the result of the aggregation obtained with the statistical median.

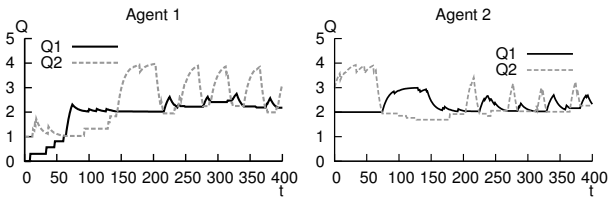

Figure 5. Example of an individual run of Q-learning in the 3rd Game.

5. Related Works

As far as we are aware, none of the existing approaches has explored the specific case of Multiagent Q-learning with ϵ-greedy exploration. The work most closely related to ours is the research of Tuyls et al. (2003). The

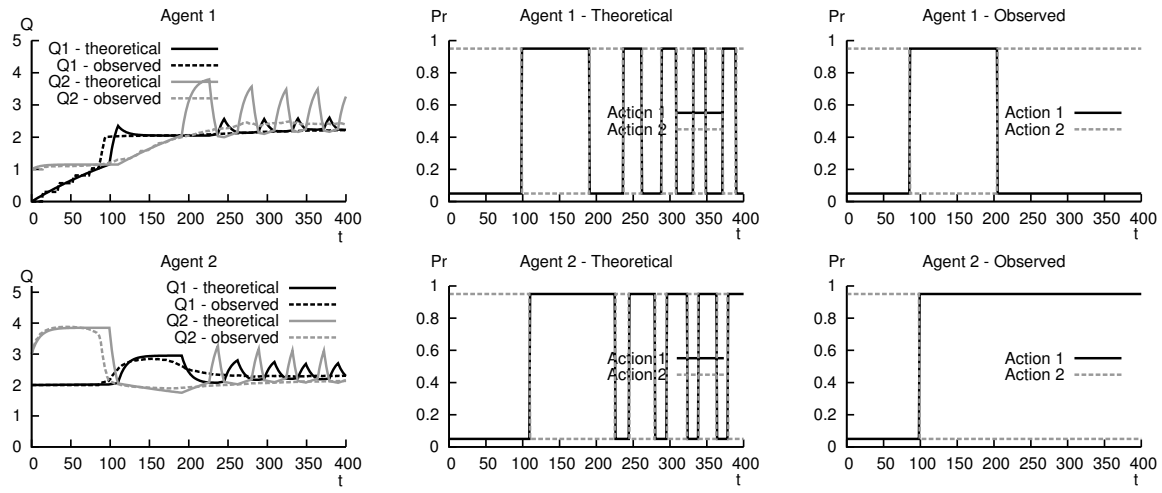

Figure 4. Graphs for the 3rd game: comparison between the theoretical Q-values derivated by the model and the median Q-values observed in the experiments (left); the expected dynamics for the agents' strategies according to the model (center); and the dynamics observed in the experiments (right).

authors have developed a continuous time model for Multiagent Q-learning with *Boltzmann* exploration and have shown a link between it and the Replicator Dynamics (RD) of Evolutionary Game Theory (Hofbauer & Sigmund, 1998). The same type of link has also been explored by Borgers and Sarin (1997) and Panait et al. (2008). The former investigated the behaviour of the agents in the Cross learning algorithm of Bush and Mosteller (1955). The later proposed and analysed a variation of the *Boltzmann*-based multiagent Q-learning to improve the cooperative behaviour of the agents.

In the RD, the probability of each action grows at a rate which is directly proportional to its performance against the others. Similar principles are applied in several Reinforcement Learning algorithms, including Q-learning with *Boltzmann* exploration. So the inspiration to develop RD-like analysis for those algorithms is quite natural. The ϵ-greedy mechanism, however, generates a different dynamics. More specifically, the mechanism defines a semi-uniform probability distribution in which the current best action is selected with probability $1-\epsilon$ and a random action with probability ϵ. Such a distribution is non-continuous and defined by a conditional function. Hence, the link cannot be directly applied in our case.

In another approach, Vidal and Durfee (2003) present a framework to track the error in one agent's decision during the multiagent learning process. The framework is generic enough to cover several different algorithms. However, it requires the tuning of some parameters that might not be known *a priori* or even impossible to obtain without extensive simulations. Our approach, in contrast, does not use any parameters other than the ones that are required by the Q-learning algorithm.

As in our research, all the above works share the property of being based on the analysis of differential or difference equations. The topic has a long research tradition in the mathematical disciplines, a considerable theoretical framework and forms the standard aproach to the study of dynamical systems. Other examples of the application of the approach to analyse multiagent reinforcement algorithms are the works of Abdallah and Lesser (2008), who applied differential equations to study the dynamics of their *Weighted Policy Learner* algorithm, and Leslie and Collins (2005), who studied the asymptotic behaviour of variants of the *Boltzmann*-based multiagent Q-learning. The approach has also been used for the analysis of single-agent reinforcement learning algorithms (Borkar & Meyn, 2000).

A different line of investigation corresponds to the identification of factors that lead the agents to develop some types of behaviours. Fulda and Ventura (2007), for example, presented a set of conditions, on the environment and the payoff tables, which are sufficient to guarantee optimal performance of cooperative agents using Q-learning with *Boltzmann* exploration. Claus and Boutilier (1998) also studied the cooperative case. Their analysis of Multiagent Q-learning with *Boltzmann* exploration has shown that the agents tend to converge to the most profitable equilibrium in simple games.

6. Conclusions

In this paper we have presented a framework to model the behaviour of Q-learning agents using the ϵ-greedy exploration mechanism. For this, we analysed a continuous-time version of the Q-learning update rule and studied how the presence of other agents and the ϵ-greedy mechanism affect it. We then modelled the problem as a system of difference equations which was used to calculate the expected evolution of the Q-values and, consequently, the expected behaviour of the agents.

The application of the model in the typical games selected from the literature has shown its feasibility. The model was able to capture all the major trends found in the experiments.

The next step in this research is to extend the approach to multi-state scenarios.

References

Abdallah, S., & Lesser, V. (2008). Non-linear Dynamics in Multiagent Reinforcement Learning Algorithms. *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)* (pp. 1321–1324). Estoril, Portugal: IFAAMAS.

Borgers, T., & Sarin, R. (1997). Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, *77*, 1–14.

Borkar, V. S., & Meyn, S. P. (2000). The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, *38*, 447–469.

Bush, R. R., & Mosteller, F. (1955). *Stochastic models for learning*. New York: John Wiley and Sons.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 746–752). Menlo Park, CA, USA: AAAI.

Fulda, N., & Ventura, D. (2007). Predicting and preventing coordination problems in cooperative Q-learning systems. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)* (pp. 780–785).

Galstyan, A., Czajkowski, K., & Lerman, K. (2004). Resource allocation in the grid using reinforcement learning. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)* (pp. 1314–1315). Washington, DC, USA: IEEE Computer Society.

Gomes, E. R., & Kowalczyk, R. (2007). Learning the ipa market with individual and social rewards. *Proceedings of the International Conference on Intelligent Agent Technology (IAT'07)* (pp. 328–334). Fremont, CA, USA: IEEE Computer Society.

Harandi, M. T., Ahmadabadi, M. N., & Araabi, B. N. (2008). Optimal local basis: A reinforcement learning approach for face recognition. *International Journal of Computer Vision*, *81*, 191–204.

Hofbauer, J., & Sigmund, K. (1998). *Evolutionary games and population dynamics*. Cambridge University Press.

Iglesias, A., Martnez, P., Aler, R., & Fernndez, F. (2008). Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence*, in press.

Leslie, D. S., & Collins, E. J. (2005). Individual Q-learning in normal form games. *SIAM Journal on Control and Optimization*, *44*, 495–514.

Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, *11*, 387–434.

Panait, L., Tuyls, K., & Luke, S. (2008). Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, *9*, 423–457.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tuyls, K., Verbeeck, K., & Lenaerts, T. (2003). A selection-mutation model for Q-learning in multi-agent systems. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)* (pp. 693–700). New York, NY, USA: ACM.

Vidal, J. M., & Durfee, E. H. (2003). Predicting the expected behavior of agents that learn about agents: the CLRI framework. *Autonomous Agents and Multi-Agent Systems*, *6*, 77–107.

Ziogos, N. P., Tellidou, A. C., Gountis, V. P., & Bakirtzis, A. G. (2007). A reinforcement learning algorithm for market participants in FTR auctions. *Proceedings of the Seventh IEEE Power Tech* (pp. 943–948). IEEE.