Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Evolutionary Dynamics of $\mathcal{Q}$–Learning over the Sequence Form

Fabio Panozzo and **Nicola Gatti** and **Marcello Restelli**
Department of Electronics, Information and Bioengineering, Politecnico di Milano
Piazza Leonardo da Vinci, 32
I-20133, Milan, Italy
{panozzo.fabio,nicola.gatti,marcello.restelli}@polimi.it

Abstract

Multi–agent learning is a challenging open task in artificial intelligence. It is known an interesting connection between multi–agent learning algorithms and evolutionary game theory, showing that the learning dynamics of some algorithms can be modeled as replicator dynamics with a mutation term. Inspired by the recent sequence–form replicator dynamics, we develop a new version of the $\mathcal{Q}$–learning algorithm working on the sequence form of an extensive–form game allowing thus an exponential reduction of the dynamics length w.r.t. those of the normal form. The dynamics of the proposed algorithm can be modeled by using the sequence–form replicator dynamics with a mutation term. We show that, although sequence–form and normal–form replicator dynamics are realization equivalent, the $\mathcal{Q}$–learning algorithm applied to the two forms have non–realization equivalent dynamics. Originally from the previous works on evolutionary game theory models form multi–agent learning, we produce an experimental evaluation to show the accuracy of the model.

Introduction

The study of *games* among *rational agents* is central in artificial intelligence. *Game theory* provides the most elegant models (Fudenberg and Tirole 1991), while *theory of algorithms* and *machine learning* (Shoham and Leyton-Brown 2009) provide the tools to design agents playing optimally. In this paper, we focus on the problem of learning optimal strategies in extensive–form games (Fudenberg and Tirole 1991). These games provide a richer representation than strategic–form games, allowing agents to play sequentially.

The problem of learning when actions are perfectly observable (aka *perfect–information* games) is well understood and a number of algorithms converging to the equilibrium are known. Instead, learning in *imperfect–information* games is a challenging open task. Some algorithms are based on the minimization of the regret. In particular, *counterfactual regret* (CFR) minimization (Zinkevich et al. 2007) produces, in self–play, average strategy profiles that converge to a Nash equilibrium in two–player zero–sum games. To scale to very large games (e.g., Poker), several researchers have focused on game abstractions and Monte–Carlo sampling techniques (Ponsen, de Jong, and Lanctot 2011). CFR has also been studied in games with imperfect recall (Lanctot et al. 2012) and it has been deeply evaluated in three–player games (Risk and Szafron 2010). In the case of general–sum games, it is only known that strictly dominated actions will be played with probability zero (Gibson 2013), but no result characterizing its dynamics is known. Some results are instead known about the characterization of the dynamics of a number of learning algorithms when applied to strategic–form games. Time-limit dynamics of the *$\mathcal{Q}$–learning* algorithm in self–play with Boltzmann exploration can be modeled, in expectation, as *replicator dynamics* with a specific mutation term (Tuyls, Hoen, and Vanschoenwinkel 2006), showing that the convergence is possible only to a Nash equilibrium when the exploration rate goes to zero. In (Wunder, Littman, and Babes 2010; Gomes and Kowalczyk 2009), the dynamics with ϵ–greedy exploration are studied. Similar results are provided for other learning algorithms (Kaisers, Bloembergen, and Tuyls 2012), including *cross learning*, *regret minimization*, and *frequency adjusted $\mathcal{Q}$–learning*. Notably, $\mathcal{Q}$–learning achieved consistently excellent results with strategic–form games, in many senses outperforming algorithms based on deeper insights about the multi–agent setting (Zawadzki, Lipson, and Leyton-Brown 2014). Although in principle these results are applicable also to the normal form of an extensive–form game, in practice they cannot be applied because the normal form is exponentially large in the size of the game tree, requiring exponentially long dynamics (and would suffer from numerical stability issues). A common alternative is to perform $\mathcal{Q}$–learning in behavioral strategies, where each information set is considered as a (partially observable) state. Some works studied the dynamics of multi–agent learning algorithms for (perfectly observable) stochastic games, showing that the dynamics are switching (Vrancx, Tuyls, and Westra 2008). Due to complexity of the dynamics, such results can be applied only to game instances with a very small number of states and their extension to imperfect–information games seems to be unfeasible even in the case of finite horizon.

Recently, in (Gatti, Panozzo, and Restelli 2013), replicator dynamics have been adapted to the *sequence form* of the extensive–form games (von Stengel 1996), allowing an exponential reduction of the size, but keeping dynamics that

are realization equivalent (i.e., they induce the same probability distribution on terminal nodes) to the dynamics with the normal form. In (Lanctot 2014), the author shows that, in the case of two–agent zero–sum games, the discrete–time sequence–form replicator dynamics have a form of counterfactual regret, converging to the Nash equilibrium.

In this paper, we provide the following contributions.
1) We introduce a novel extension of the $\mathcal{Q}$–learning algorithm that operates on the sequence form.
2) We show that the learning dynamics of the proposed sequence–form $\mathcal{Q}$–learning algorithm can be modeled by the sequence–form replicator dynamics (Gatti, Panozzo, and Restelli 2013), presenting exponentially smaller dynamics w.r.t. those of (Tuyls, Hoen, and Vanschoenwinkel 2006). We leave open the problem to extend such an approach to other learning algorithms, e.g., those in (Kaisers, Bloembergen, and Tuyls 2012).
3) We show that, although sequence–form and normal–form replicator dynamics are realization–equivalent, the dynamics of our $\mathcal{Q}$–learning algorithm are not realization equivalent to the ones of (Tuyls, Hoen, and Vanschoenwinkel 2006), due to the mutation terms, and we analyze the accuracy of our model w.r.t. the actual dynamics of the algorithm.

Preliminaries

Extensive–form games. A *perfect–information* extensive–form game (Fudenberg and Tirole 1991) is a tuple $(N, A, V, T, \iota, \rho, \chi, \mathbf{u})$, where: N is the set of agents ($i \in N$ denotes a generic agent), A is the set of actions ($A_i \subseteq A$ denotes the set of actions of agent i and $a \in A$ denotes a generic action), V is the set of decision nodes ($V_i \subseteq V$ denotes the set of decision nodes of i), T is the set of terminal nodes ($w \in V \cup T$ denotes a generic node and w_0 is root node), $\iota : V \to N$ returns the agent that acts at a given decision node, $\rho : V \to \wp(A)$ returns the actions available to agent $\iota(w)$ at w, $\chi : V \times A \to V \cup T$ assigns the next (decision or terminal) node to each pair $\langle w, a\rangle$ where a is available at w, and $\mathbf{u} = (u_1, \ldots, u_{|N|})$ is the set of agents' utility functions $u_i : T \to \mathbb{R}$. Games with *imperfect information* extend those with perfect information, allowing one to capture situations in which some agents cannot observe some actions undertaken by other agents. We denote by $V_{i,h}$ the h–th *information set* of agent i. An information set is a set of decision nodes such that when an agent plays at one of such nodes she cannot distinguish the node in which she is playing. For the sake of simplicity, we assume that every information set has a different index h, thus we can univocally identify an information set by h. Furthermore, since the available actions at all nodes w belonging to the same information set h are the same, with abuse of notation, we write $\rho(h)$ in place of $\rho(w)$ with $w \in V_{i,h}$. An imperfect–information game is a tuple $(N, A, V, T, \iota, \rho, \chi, \mathbf{u}, H)$ where $(N, A, V, T, \iota, \rho, \chi, \mathbf{u})$ is a perfect–information game and $H = (H_1, \ldots, H_{|N|})$ induces a partition $V_i = \bigcup_{h \in H_i} V_{i,h}$ such that for all $w, w' \in V_{i,h}$ we have $\rho(w) = \rho(w')$. We focus on games with *perfect recall* where each agent recalls all her own previous actions and the ones of her opponents (Fudenberg and Tirole 1991).

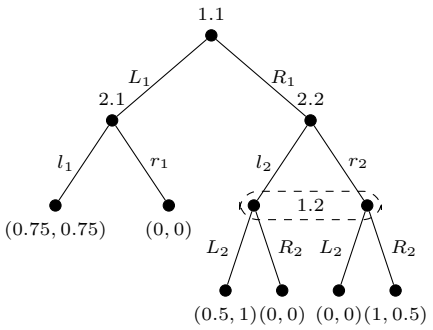

Figure 1: Example of two–agent imperfect–information extensive–form game, $\mathbf{x}.\mathbf{y}$ denotes the $\mathbf{y}$–th node of agent $\mathbf{x}$.

Sequence form (von Stengel 1996). It is a representation composed by a table and a set of constraints. Sequence–form actions are called *sequences*. A sequence $q \in Q_i$ of agent i is a set of consecutive actions $a \in A_i$ where $Q_i \subseteq Q$ is the set of sequences of agent i and Q is the set of all the sequences. A sequence can be *terminal*, if, combined with some sequence of the opponents, it leads to a terminal node, or *non–terminal* otherwise. In addition, the initial sequence of every agent, denoted by $\mathtt{q}_\emptyset$, is said *empty sequence* and, given sequence $q \in Q_i$ leading to some information set $h \in H_i$, we say that q' *extends* q and we denote by $q' = q|a$ if the last action of q' (denoted by $a(q') = a$) is some action $a \in \rho(h)$ and q leads to h. We denote by $w = h(q)$ the node w with $a(q) \in \rho(w)$; by $q' \subseteq q$ a subsequence of q; by $\mathbf{x}_i$ the sequence–form strategy of agent i and by $x_i(q)$ the probability associated with sequence $q \in Q_i$. Finally, condition $q \to h$ is true if sequence q crosses information set h. Well–defined strategies are such that, for every information set $h \in H_i$, the probability $x_i(q)$ assigned to the sequence q leading to h is equal to the sum of the probabilities $x_i(q')$s where q' extends q at h. Sequence form constraints are $x_i(\mathtt{q}_\emptyset) = 1$ and $x_i(q) = \sum_{a \in \rho(w)} x_i(q|a)$ for every sequence q, action a, node w such that $w = h(q|a)$, and for every agent i. We denote by $\mathbf{x}_i$ the strategy profile of agent i and by $\hat{\mathbf{x}}_i$ the pure strategy profiles such that $\hat{x}_i(q) = 1$ if q is played and $\hat{x}_i(q) = 0$ otherwise. Agent i's utility is represented as a sparse multi–dimensional array, denoted, with an abuse of notation, by U_i, specifying the value associated with every combination of terminal sequences of all the agents. The size of the sequence form is linear in the size of the game tree and therefore it is exponentially smaller than the normal form.

Replicator dynamics. The sequence–form replicator dynamics have been introduced in (Gatti, Panozzo, and Restelli 2013) and it has been shown to be realization equivalent to the standard replicator dynamics applied to the normal form. The sequence–form continuous–time replicator equation is:

$$\dot{x}_1(q,t) = x_1(q,t) \cdot [(\mathbf{g}_q(\mathbf{x}_1(t)) - \mathbf{x}_1(t))^T \cdot U_1 \cdot \mathbf{x}_2(t)]$$
$$\dot{x}_2(q,t) = x_2(q,t) \cdot [\mathbf{x}_1^T(t) \cdot U_2 \cdot (\mathbf{g}_q(\mathbf{x}_2(t)) - \mathbf{x}_2(t))],$$

where $\mathbf{g}_q(\mathbf{x}_i(t))$ is computed by Algorithm 1.

Algorithm 1 generate_$\mathbf{g_q}(\mathbf{x}_i(t))$

1: $\mathbf{g}_q(\mathbf{x}_i(t)) = \mathbf{0}$
2: **if** $x_i(q,t) \neq 0$ **then**
3: **for** $q' \in Q_i$ s.t. $q' \subseteq q$ **do**
4: $g_q(q', \mathbf{x}_i(t)) = 1$
5: **for** $q'' \in Q_i$ s.t. $q'' \cap q = q'$ **and** $q'' = q'|a| \ldots : a \in \rho(h), q \not\to h$ **do**
6: $g_q(q'', \mathbf{x}_i(t)) = \frac{x_i(q'',t)}{x_i(q',t)}$
7: **return** $\mathbf{g}_q(\mathbf{x}_i(t))$

Algorithm 2 SFpure_strategy$(\mathbf{x}_i)$

1: $\hat{\mathbf{x}}_i = \mathbf{0}$
2: $\hat{x}_i(\mathsf{q}_\emptyset) = 1$
3: $inf\_sets\_to\_evaluate = \{h : \exists a \in \rho(h), \mathsf{q}_\emptyset | a \in Q_i\}$
4: **while** $inf\_sets\_to\_evaluate \neq \emptyset$ **do**
5: choose an information set $h \in inf\_sets\_to\_evaluate$
6: choose a sequence q such that $a(q) \in \rho(h)$ according $\mathbf{x}_i$
7: $\hat{x}_i(q) = 1$
8: $inf\_sets\_to\_evaluate = inf\_sets\_to\_evaluate \cup \{h' : \exists a \in \rho(h'), q|a \in Q_i\}$
9: $inf\_sets\_to\_evaluate = inf\_sets\_to\_evaluate \setminus \{h\}$
10: **return** $\hat{\mathbf{x}}_i$

$\mathcal{Q}$–learning (Watkins and Dayan 1992). It is an algorithm used by learning agents in Markovian domains that allows them to learn the optimal policy without having prior knowledge about the domain dynamics and the reward function. $\mathcal{Q}$–learning is an adaptive model–free value–iteration method whose aim is to estimate the value $\mathcal{Q}_{t+1}(s,a)$ of each action a in each state s at time $t+1$ given the estimations of $\mathcal{Q}_t(s',a')$, where s' is any state reached by taking action a in s and a' is the greedy action in s', through the following rule:

$$\mathcal{Q}_{t+1}(s,a) = (1-\alpha)\mathcal{Q}_t(s,a) + \alpha\left(r(s,a) + \gamma \max_{a'} \mathcal{Q}_t(s',a')\right),$$

where $r(s,a)$ is the immediate reward received being in state s by taking action a, α is the learning rate and γ is the discount factor (Sutton and Barto 1998).

Q–learning and sequence form

The application of the $\mathcal{Q}$–learning algorithm to the normal form of an extensive–form game is a straightforward extension of (Tuyls, Hoen, and Vanschoenwinkel 2006) as well as the derivation of its time–limit (standard) replicator dynamics based model. However, the normal form being exponentially large in the size of the game tree, the learning times are exponentially long (as confirmed in our experimental results). Here, we propose a variation of the $\mathcal{Q}$–learning algorithm for the sequence form of an extensive–form game, reducing exponentially the length of the learning times. Differently from the works studying evolutionary game theory models for multi–agent learning algorithms, but consistently with the online RL literature, we consider the learning rate α and the exploitation parameter τ as functions of time.

The basic idea behind our algorithm is that each agent, at each repetition of the game, without any knowledge about the other agents' strategies, chooses a sequence–form pure strategy. Since the set of these strategies is exponentially large in the size of the game (there is one sequence–form pure strategy per plan of the reduced normal form), each agent implicitly chooses a strategy by using a procedure that, given a strategy $\mathbf{x}_i$, builds a sequence–form pure strategy $\hat{\mathbf{x}}_i$ in polynomial time w.r.t. the size of the game. The procedure, summarized in Algorithm 2, is iterative and works as follows. In Steps 1–3, the empty sequence of an agent is chosen and the information sets directly reachable by the empty sequence are inserted in the set $inf\_sets\_to\_evaluate$. Then, until such set is not empty, in Step 5 the procedure extracts an information set h, in Step 6 it randomly draws a sequence q with $a(q) \in \rho(h)$ according to strategy $\mathbf{x}_i$ once normalized by the probability to reach h and assigns probability one to q in $\hat{\mathbf{x}}_i$, in Step 8 adds all the information sets directly reachable by q to $inf\_sets\_to\_evaluate$ and in Step 9 removes h from such set. Given that each information set is evaluated no more than once, the complexity of the procedure is linear in the size of the game tree.

In our learning algorithm, we associate a $\mathcal{Q}$–value with each sequence and we update the $\mathcal{Q}$–values according to the observed outcomes. Given that each agent plays a number of sequences at each repetition of the game (a sequence–form pure strategy includes multiple sequences), we need to modify the updating rule of $\mathcal{Q}_t$ allowing an agent to update multiple $\mathcal{Q}$–values (those corresponding to the sequences of the pure–strategy profiles $\hat{\mathbf{x}}$ chosen by Algorithm 2). Formally, for each $q|a$ such that $\hat{x}_i(q|a) = 1$ we have:

$$\mathcal{Q}_{t+1}(q|a) = (1-\alpha_t)\mathcal{Q}_t(q|a) + \alpha_t\Big(r(q|a) + \gamma \max_{\substack{a' \in \rho(h):\\ a \in \rho(h)}} \mathcal{Q}_t(q|a')\Big), \tag{1}$$

where $r(q|a) = \hat{\mathbf{x}}_i^T(t) \cdot U_i \cdot \hat{\mathbf{x}}_{-i}(t)$ is the immediate reward obtained by agent i at iteration t.
The action–selection strategy is based on the Boltzmann distribution. More precisely, given the $\mathcal{Q}$–value of each sequence, we derive a sequence–form strategy profile as:

$$x_i(q|a,t) = x_i(q,t) \cdot \frac{e^{\tau_t \mathcal{Q}_t(q|a)}}{\sum_{a' \in \rho(h): a \in \rho(h)} e^{\tau_t \mathcal{Q}_t(q|a')}}. \tag{2}$$

The learning algorithm is summarized in Algorithm 3.

In Steps 2–3, the algorithm initializes the $\mathcal{Q}$–value associated with each sequence of agent i to a random value. In Steps 4–5, the algorithm derives the strategy profile specifying the probability distribution over the sequences by Boltzmann equation on the basis of the $\mathcal{Q}$–values. In Step 6, the algorithm repeats the following steps until agent i has learned the optimal policy. In Steps 7–8, agent i draws a sequence–form pure strategy as prescribed by Algorithm 2. In Steps 9–10, for each sequence chosen by Algorithm 2, the algorithm updates the $\mathcal{Q}$–values applying the updating rule Eq. (1). In Steps 11–12, the algorithm updates the probability distribution over the sequences given the new $\mathcal{Q}$–values. The algorithm is repeated until the maximum variation of the $\mathcal{Q}$–values is less than a given threshold ϵ.

Algorithm 3 SF$\mathcal{Q}$_learning

$t = 0$
for all $q \in Q_i$ **do**
  $\mathcal{Q}_t(q)$ drawn uniformly from $\left[\frac{\min_{\mathbf{x}} U_i(\mathbf{x})}{1-\gamma}, \frac{\max_{\mathbf{x}} U_i(\mathbf{x})}{1-\gamma}\right]$
for all $q|a \in Q_i$ **do**
  $x_i(q|a,t) = x_i(q,t) \cdot \frac{e^{\tau_t \mathcal{Q}_t(q|a)}}{\sum_{a' \in \rho(h): a \in \rho(h)} e^{\tau_t \mathcal{Q}_t(q|a')}}$
repeat
  $\hat{\mathbf{x}}_i = \text{sequence\_form\_pure\_strategy}(\mathbf{x}_i)$
  play $\hat{\mathbf{x}}_i$
  for all $q|a$ such that $\hat{x}_i(q|a) == 1$ **do**
    $\mathcal{Q}_{t+1}(q|a) = (1-\alpha_t)\mathcal{Q}_t(q|a) + \alpha_t(r(q|a) + \gamma \max_{a' \in \rho(h): a \in \rho(h)} \mathcal{Q}_t(q|a'))$
  for all $q|a \in Q_i$ **do**
    $x_i(q|a,t) = x_i(q,t) \cdot \frac{e^{\tau_t \mathcal{Q}_t(q|a)}}{\sum_{a' \in \rho(h): a \in \rho(h)} e^{\tau_t \mathcal{Q}_t(q|a')}}$
  $t = t + 1$
until $\max_{q \in Q_i} |\mathcal{Q}_{t+1}(q) - \mathcal{Q}_t(q)| < \epsilon$
return $\mathbf{x}$

Dynamical analysis

We study the dynamics of Algorithm 3 in expectation w.r.t. its realizations and their relationship with the replicator dynamics when there are 2 agents. The generalization with more agents is straightforward. We assume, as in (Tuyls, Hoen, and Vanschoenwinkel 2006), that the game is continuously repeated, the time between a repetition and the subsequent one tending to zero. This allows us to calculate (in expectation) the derivative of Eq. (2) as:

$$\dot{x}_i(q|a,t) = \dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)} + x_i(q|a,t)\Big(\dot{\tau}_t \mathcal{Q}_t(q|a) + \tau_t \dot{\mathcal{Q}}_t(q|a) - \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)}\Big(\dot{\tau}_t \mathcal{Q}_t(q|a') + \tau_t \dot{\mathcal{Q}}_t(q|a')\Big)\Big) \quad (3)$$

and the time derivative of Eq. (1) as

$$\dot{\mathcal{Q}}_t(q|a) = \alpha_t \left(r(q|a) + \gamma \max_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \mathcal{Q}_t(q|a') - \mathcal{Q}_t(q|a) \right). \quad (4)$$

By replacing $\dot{\mathcal{Q}}_t(q|a)$ in Eq. (3) with the regret term of Eq. (4) we obtain

$$\dot{x}_i(q|a,t) = \dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)} + x_i(q|a,t)\alpha_t \Big(\dot{\tau}_t \mathcal{Q}_t(q|a) + \tau_t \Big(r(q|a) + \gamma \max_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \mathcal{Q}_t(q|a') - \mathcal{Q}_t(q|a)\Big) - \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \cdot \Big(\dot{\tau}_t \mathcal{Q}_t(q|a') + \tau_t \Big(r(q|a') + \gamma \max_{\substack{a'' \in \rho(h): \\ a' \in \rho(h)}} \mathcal{Q}_t(q|a'') - \mathcal{Q}_t(q|a')\Big)\Big)\Big).$$

Given that $\sum_{a' \in \rho(h): a \in \rho(h)} x_i(q|a',t) = x_i(q,t)$ by definition of sequence form, we have that

$$\max_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \mathcal{Q}_t(q|a') - \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \gamma \max_{\substack{a'' \in \rho(h): \\ a' \in \rho(h)}} \mathcal{Q}_t(q|a'') = 0$$

thus, we can rewrite

$$\dot{x}_i(q|a,t) = \dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)} + x_i(q|a,t)\alpha_t \Big(\dot{\tau}_t \mathcal{Q}_t(q|a) + \tau_t \Big(r(q|a) - \mathcal{Q}_t(q|a)\Big) - \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \Big(\dot{\tau}_t \mathcal{Q}_t(q|a') + \tau_t \Big(r(q|a') - \mathcal{Q}_t(q|a')\Big)\Big)\Big). \quad (5)$$

We can rewrite the reward in expectation of a sequence as its fitness as

$$r(q|a) = \mathbf{g}_{q|a}^T(\mathbf{x}_i(t)) \cdot U_i \cdot \mathbf{x}_{-i}(t)$$

thus Eq. (5) becomes

$$\dot{x}_i(q|a,t) = \dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)} + x_i(q|a,t)\alpha_t \left(\dot{\tau}_t \mathcal{Q}_t(q|a) + \tau_t \left(\mathbf{g}_{q|a}^T(\mathbf{x}_i(t)) \cdot U_i \cdot \mathbf{x}_{-i}(t) - \mathcal{Q}_t(q|a)\right) - \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \cdot \left(\dot{\tau}_t \mathcal{Q}_t(q|a') + \tau_t \left(\mathbf{g}_{q|a'}^T\left(\mathbf{x}_i(t)\right) \cdot U_i \cdot \mathbf{x}_{-i}(t) - \mathcal{Q}_t(q|a')\right)\right)\right).$$

Given that

$$\sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \mathbf{g}_{q|a'}^T(\mathbf{x}_i(t)) \cdot U_i \cdot \mathbf{x}_{-i}(t) = \mathbf{g}_q^T(\mathbf{x}_i(t)) \cdot U_i \cdot \mathbf{x}_{-i}(t)$$

we have

$$\dot{x}_i(q|a,t) = \dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)} + x_i(q|a,t)\alpha_t \cdot \left(\tau_t \left(\left(\mathbf{g}_{q|a}(\mathbf{x}_i(t)) - \mathbf{g}_q(\mathbf{x}_i(t))\right)^T \cdot U_i \cdot \mathbf{x}_{-i}(t)\right) - \tau_t \mathcal{Q}_t(q|a) + \dot{\tau}_t \mathcal{Q}_t(q|a) - \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \left(\dot{\tau}_t \mathcal{Q}_t(q|a') - \tau_t \mathcal{Q}_t(q|a')\right)\right).$$

By Boltzmann distribution, for every q, a, a', we have $\frac{x_i(q|a',t)}{x_i(q|a,t)} = \frac{e^{\tau_t \mathcal{Q}_t(q|a')}}{e^{\tau_t \mathcal{Q}_t(q|a)}}$ and by sequence form definition we have $\sum_{a' \in \rho(h): a \in \rho(h)} \frac{x_i(q|a',t)}{x_i(q,t)} = 1$ for every q, so we obtain:

$$
\begin{aligned}
&- \tau_t \mathcal{Q}_t(q|a) + \tau_t \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \mathcal{Q}_t(q|a') \\
&= \tau_t \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} (\mathcal{Q}_t(q|a') - \mathcal{Q}_t(q|a)) \\
&= \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \log \left(\frac{x_i(q|a',t)}{x_i(q|a,t)} \right).
\end{aligned}
$$

Thus

$$
\begin{aligned}
\dot{x}_i(q|a,t) = x_i(q|a,t) \Bigg[&\alpha_t \tau_t \left((\mathbf{g}_{q|a}(\mathbf{x}_i(t)) - \mathbf{x}_i(t))^T \cdot U_i \cdot \mathbf{x}_{-i}(t) \right) \\
&+ \left(\alpha_t + \frac{\dot{\tau}_t}{\tau_t} \right) \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(q,t)} \log \left(\frac{x_i(q|a',t)}{x_i(q|a,t)} \right) \Bigg] + \\
&\dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)}. \quad (6)
\end{aligned}
$$

When $q = \mathsf{q}_\emptyset$, the time derivative $\dot{x}_i(\mathsf{q}_\emptyset, t) = 0$ because, by definition of sequence form, $x_i(\mathsf{q}_\emptyset, t) = 1$ for every t. Thus, when $q = \mathsf{q}_\emptyset$ Eq. (6) becomes:

$$
\begin{aligned}
\dot{x}_i(\mathsf{q}_\emptyset|a,t) = x_i(\mathsf{q}_\emptyset|a,t) \Bigg[&\alpha_t \tau_t \left((\mathbf{g}_{\mathsf{q}_\emptyset|a}(\mathbf{x}_i(t)) - \mathbf{x}_i(t))^T \cdot U_i \cdot \mathbf{x}_{-i}(t) \right) \\
&+ \left(\alpha_t + \frac{\dot{\tau}_t}{\tau_t} \right) \sum_{\substack{a' \in \rho(h): \\ a \in \rho(h)}} \frac{x_i(q|a',t)}{x_i(\mathsf{q}_\emptyset,t)} \log \left(\frac{x_i(\mathsf{q}_\emptyset|a',t)}{x_i(\mathsf{q}_\emptyset|a,t)} \right) \Bigg]. \quad (7)
\end{aligned}
$$

By substituting iteratively $\dot{x}_i(\mathsf{q}_\emptyset|a,t)$ in the term $\dot{x}_i(q,t) \cdot \frac{x_i(q|a,t)}{x_i(q,t)}$ of Eq. (6) for every q, we obtain:

$$
\begin{aligned}
\dot{x}_1(q|a,t) = x_1(q|a,t) \Bigg[&\alpha_t \tau_t \left((\mathbf{g}_{q|a}(\mathbf{x}_1(t)) - \mathbf{x}_1(t))^T \cdot U_1 \cdot \mathbf{x}_2(t) \right) \\
&+ \left(\alpha_t + \frac{\dot{\tau}_t}{\tau_t} \right) \sum_{\substack{h: \exists a^* \in q|a \\ a^* \in \rho(h)}} \sum_{a' \in \rho(h)} \frac{x_1(q|a',t)}{x_1(q,t)} \log \left(\frac{x_1(q|a',t)}{x_1(q|a^*,t)} \right) \Bigg] \quad (8)
\end{aligned}
$$

$$
\begin{aligned}
\dot{x}_2(q|a,t) = x_2(q|a,t) \Bigg[&\alpha_t \tau_t \left(\mathbf{x}_1(t)^T \cdot U_2 \cdot (\mathbf{g}_{q|a}(\mathbf{x}_2(t)) - \mathbf{x}_2(t)) \right) \\
&+ \left(\alpha_t + \frac{\dot{\tau}_t}{\tau_t} \right) \sum_{\substack{h: \exists a^* \in q|a \\ a^* \in \rho(h)}} \sum_{a' \in \rho(h)} \frac{x_2(q|a',t)}{x_2(q,t)} \log \left(\frac{x_2(q|a',t)}{x_2(q|a^*,t)} \right) \Bigg]. \quad (9)
\end{aligned}
$$

The replicator dynamics Eq. (8)–(9) is formed by two terms: the selection (exploitation) one,

$$
x_i(q|a,t) \alpha_t \tau_t \left((\mathbf{g}_{q|a}(\mathbf{x}_i(t)) - \mathbf{x}_i(t))^T \cdot U_i \cdot \mathbf{x}_{-i}(t) \right)
$$

and the mutation (exploration) one,

$$
x_i(q|a,t) \left(\alpha_t + \frac{\dot{\tau}_t}{\tau_t} \right) \sum_{\substack{h: \exists a^* \in q|a \\ a^* \in \rho(h)}} \sum_{\substack{a' \in \rho(h), \\ q' \subseteq q}} \frac{x_i(q'|a',t)}{x_i(q,t)} \log \frac{x_i(q'|a',t)}{x_i(q'|a^*,t)},
$$

where $q' \subseteq q$ is the sequence leading to h. The learning rate α_t (contained in both terms) has the function of changing the speed of dynamics, but it does not affect the trajectory. More interesting is the exploitation parameter τ_t whose function is to increase or decrease the prominence of the selection term w.r.t. the mutation term. When τ_t increases (decreases), Algorithm 3 assigns a greater (smaller) probability to sequences with high $\mathcal{Q}$–values, preferring the exploitation (exploration) w.r.t. the exploration (exploitation); the same reflects in the replicator dynamics because a greater (smaller) τ_t means that the selection term affects the learning dynamics more (less) than the mutation one.

By exploiting the same arguments discussed in (Gatti, Panozzo, and Restelli 2013), it can be observed that the selection term of replicator dynamics in Eq. (8)–(9) is realization equivalent to the selection term of the replicator dynamics studied in (Tuyls, Hoen, and Vanschoenwinkel 2006). That is, the learning dynamics in expectation of our $\mathcal{Q}$–learning algorithm are realization equivalent to the learning dynamics of the model proposed in (Tuyls, Hoen, and Vanschoenwinkel 2006) once applied to the normal form when the mutation term is zero. When the mutation term tends to zero, the two replicator dynamics models have the same rest points, but they can have different trajectories. When instead the mutation term does not tend to zero, the two replicator dynamics have different trajectories and rest points. An example is reported in Fig. 2(a).

Theorem 1 *Given*

- *a normal–form strategy profile* $(\boldsymbol{\pi}_1(t), \boldsymbol{\pi}_2(t))$ *and its evolution* $(\boldsymbol{\pi}_1(t + \Delta t), \boldsymbol{\pi}_2(t + \Delta t))$ *according to (Tuyls, Hoen, and Vanschoenwinkel 2006),*
- *a sequence–form strategy profile* $(\mathbf{x}_1(t), \mathbf{x}_2(t))$ *and its evolution* $(\mathbf{x}_1(t + \Delta t), \mathbf{x}_2(t + \Delta t))$ *according to (8)–(9),*

if $(\boldsymbol{\pi}_1(t), \boldsymbol{\pi}_2(t))$ *and* $(\mathbf{x}_1(t), \mathbf{x}_2(t))$ *are realization equivalent, in general* $(\boldsymbol{\pi}_1(t + \Delta t), \boldsymbol{\pi}_2(t + \Delta t))$ *and* $(\mathbf{x}_1(t + \Delta t), \mathbf{x}_2(t + \Delta t))$ *are not realization equivalent.*

Since the selection terms are realization equivalent, the non–equivalence is due to the mutation terms. More precisely, while the mutation term associated with a normal–form action depends on all the strategies over all the other actions (Tuyls, Hoen, and Vanschoenwinkel 2006), the mutation term associated with a sequence q depends on the strategies over a subset of sequences (i.e., all the sequences that can be played at every information set crossed by q).

Experimental results

Relationship between replicators dynamics. As shown by Theorem 1, the replicator dynamics (8)–(9) and the one described in (Tuyls, Hoen, and Vanschoenwinkel 2006) are not realization equivalent, but only their selection terms are. In particular, when payoffs are normalized in $[0, 1]$ such as in Fig. 1, we observed that at $\tau = 5$ the two trajectories are very far, while as τ approaches a value of 15 the two trajectories get close. Examples of trajectories of the two replicators, starting from the same point, for increasing values of τ are shown in (Panozzo, Gatti, and Restelli 2014).

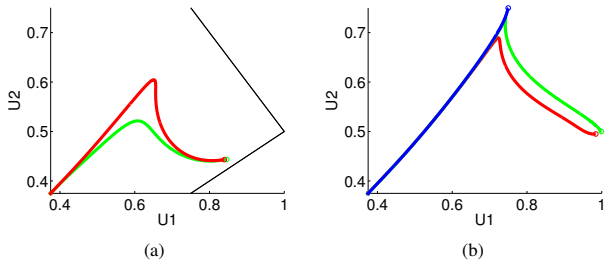

Figure 2: (a) Sequence–form (red line) and normal–form (green line) replicator dynamics in the running example of Fig. 1 with $\alpha = 1$ and $\tau = 5$. (b) Example of different limit points of learning dynamics from the same starting point with constant but different τ; red line $\tau = 10$, green line $\tau = 20$, blue line $\tau = 30$.

Learning dynamics length. We compare the learning dynamics length (in terms of iterations) obtained when the $\mathcal{Q}$–learning is applied to the normal form w.r.t. our sequence form version when the initial strategies are realization equivalent and with different configurations of the parameters as the size of the tree (branching factor b and depth d) varies. In our experimental setting τ is a linear increasing function of time starting from 0.0001 and ending to 0.5, α is exponential decreasing starting from 1 and ending to 0.2. The algorithm stops when the difference of expected utility between iterations n and $n-1$ of both agents is smaller than 0.001 for 1000 consecutive iterations. However, the average length is comparable only for very small game trees: with $b = 2$ and $d = 2$ the normal form requires about 1.5 times the number of iterations required by the sequence form, while with $b = 2$ and $d = 3$ the ratio is about 2.7; with larger d the ratio is larger than 1000. This confirms that the dynamics in normal form are exponentially longer than those in the sequence form and this is because the normal form is exponentially large w.r.t. the sequence form.

Parameters influence in the learning limit points. Parameters α and τ affect in different ways the trajectories of (8)–(9). While α only affects the dynamics speed without changing the trajectory, τ increases or decreases the prominence of selection term w.r.t. the mutation term. The parameter τ can dramatically affect the trajectories of the replicator dynamics changing both the learning limit points (by introducing small perturbations) and their basins of attraction. An example is shown in Fig. 2(b). Increasing τ from 10 to 20 the learning limit point is slightly perturbed getting close to the equilibrium with utilities $(1, 0.5)$, while increasing τ from 20 to 30 makes the basins of attraction change and the learning trajectory converges to a different limit point.

Variability analysis. The replicator dynamics model provides an abstract (time–limit) description in expectation of the learning dynamics. We evaluate here the accuracy of the model w.r.t. the execution of the algorithm. We executed 100 times our algorithm from the same starting point for each different combination of parameters $\alpha \in \{0.01, 0.05, 0.1\}$ and $\tau \in \{5, 9\}$. For each parameter combination, we measured the distance between each trajectory and the trajectory

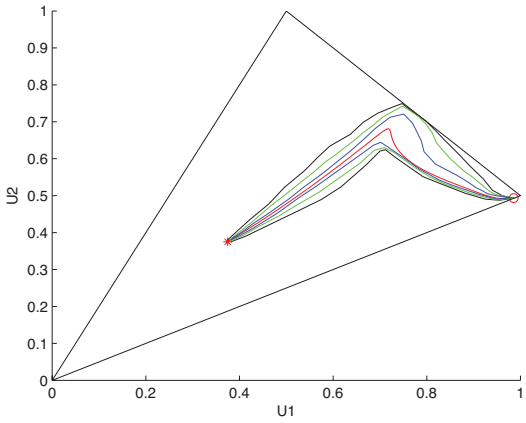

Figure 3: Statistical bounds that represent the distance between the expected (from replicator dynamics) learning dynamics (red line) and the real dynamics of learning agents that use our algorithm (black line 75%, green line 50%, blue line 25%) with $\tau = 9$ and $\alpha = 0.01$.

prescribed by the replicator dynamics model. In Fig. 3 we reported, with $\alpha = 0.01$ and $\tau = 9$, the replicator dynamics trajectory (red curve) and the curves containing the 25 closest dynamics (blue curve), the 50 ones (green curve) and the 75 ones (black curve) in the utility space (in the strategy space the graphical representation is not possible). Curves for different values of α and τ can be found in (Panozzo, Gatti, and Restelli 2014). Interestingly, the learning trajectories converge in a neighborhood of the limit point of the replicator dynamics independently of the value of τ. The specific value of τ affects the possibility to converge to (a neighborhood of) an equilibrium. Instead, the variance of the learning process is strictly related to the value of α. If α is very small, then the variance is small and, as α goes to zero, the learning process is well approximated by our model.

Conclusion and future works

We developed an efficient $\mathcal{Q}$–learning based algorithm that works with the sequence form representation of extensive–form games. We showed that the time–limit learning dynamics of our algorithm in expectation can be described by means of a new sequence–form replicator dynamics with a mutation term. Finally, we experimentally evaluated our algorithm applied to the sequence form and we showed the improvement w.r.t. the $\mathcal{Q}$–learning algorithm applied to the normal form and an analysis about how far the actual trajectories can depart from the replicator dynamic model.

In future, we intend to explore the following problems: defining a $\mathcal{Q}$–learning based algorithm working with the agent form representation of extensive–form games, applying our algorithm to state–action model of extensive–form games, deriving theoretical bounds in probability (e.g., Hoeffding's and Chernoff's) over the distance between the trajectories predicted by the model and the actual trajectories, and extend our work to other learning algorithms.

References

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. MIT Press, Cambridge, MA.

Gatti, N.; Panozzo, F.; and Restelli, M. 2013. Efficient evolutionary dynamics with extensive–form games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 335–341.

Gibson, R. 2013. Regret minimization in non-zero-sum games with applications to building champion multiplayer computer poker agents. *arXiv preprint arXiv:1305.0034*.

Gomes, E. R., and Kowalczyk, R. 2009. Dynamic analysis of multiagent q–learning with ϵ–greedy exploration. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 369–376. ACM.

Kaisers, M.; Bloembergen, D.; and Tuyls, K. 2012. A common gradient in multi–agent reinforcement learning. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, 1393–1394. International Foundation for Autonomous Agents and Multiagent Systems.

Lanctot, M.; Gibson, R.; Burch, N.; Zinkevich, M.; and Bowling, M. 2012. No-regret learning in extensive-form games with imperfect recall. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning (ICML 2012)*.

Lanctot, M. 2014. Further developments of extensive–form replicator dynamics using sequence-form representations. In *Proceedings of the Thirteenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Panozzo, F.; Gatti, N.; and Restelli, M. 2014. Evolutionary dynamics of Q–learning over the sequence form. *arXiv*.

Ponsen, M.; de Jong, S.; and Lanctot, M. 2011. Computing approximate nash equilibria and robust best-responses using sampling. *Journal of Artificial Intelligence Research* 42(1):575–605.

Risk, N. A., and Szafron, D. 2010. Using counterfactual regret minimization to create competitive multiplayer poker agents. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 159–166. International Foundation for Autonomous Agents and Multiagent Systems.

Shoham, Y., and Leyton-Brown, K. 2009. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, USA: MIT Press.

Tuyls, K.; Hoen, P. J.; and Vanschoenwinkel, B. 2006. An evolutionary dynamical analysis of multi–agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems* 12(1):115–153.

von Stengel, B. 1996. Efficient computation of behavior strategies. *Games and Economic Behavior* 14(2):220–246.

Vrancx, P.; Tuyls, K.; and Westra, R. L. 2008. Switching dynamics of multi–agent learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, 307–313. International Foundation for Autonomous Agents and Multiagent Systems.

Watkins, C. J., and Dayan, P. 1992. Q–learning. *Machine learning* 8(3–4):279–292.

Wunder, M.; Littman, M. L.; and Babes, M. 2010. Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 1167–1174.

Zawadzki, E.; Lipson, A.; and Leyton-Brown, K. 2014. Empirically Evaluating Multiagent Learning Algorithms. *arXiv:1401.8074*.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20*, 1729–1736.