International Journal of Knowledge-based and Intelligent Engineering Systems 15 (2011) 55–64
DOI 10.3233/KES-2010-0206
IOS Press

The world of independent learners is not markovian

Guillaume J. Laurent*, Laëtitia Matignon and N. Le Fort-Piat
FEMTO-ST Institute, CNRS / ENSMM / UFC/ UTBM, 24 rue Alain Savary, 25000 Besançon, France

Abstract. In multi-agent systems, the presence of learning agents can cause the environment to be non-Markovian from an agent's perspective thus violating the property that traditional single-agent learning methods rely upon. This paper formalizes some known intuition about concurrently learning agents by providing formal conditions that make the environment non-Markovian from an independent (non-communicative) learner's perspective. New concepts are introduced like the divergent learning paths and the observability of the effects of others' actions. To illustrate the formal concepts, a case study is also presented. These findings are significant because they both help to understand failures and successes of existing learning algorithms as well as being suggestive for future work.

Keywords: Multi-agent system, machine learning, reinforcement learning

1. Introduction

The problem of an agent learning to act in an unknown world is both challenging and interesting. Besides challenges inherited from single-agent learning, including the curse of dimensionality, the exploration-exploitation trade-off and the issue of observability of state, several new challenges arise in multi-agent learning [31]. Foremost among these is the difficulty of defining good learning goals for the multiple learning agents [6]. Furthermore, in the case of cooperative tasks, a learner must coordinate its behavior with other learners, such that a coherent joint behavior results [11]. Finally, the presence of multiple concurrent learners makes the environment non-stationary from a single agent's perspective. The loss of environmental stationarity is a commonly cited cause of difficulties in multi-agent learning systems [5,10,24,26,35].

A process is said non-stationary if its transition probabilities change with the time. A non-stationary process can be Markovian if the evolution of its transition probabilities depends only on time and not on the history of actions and states. Most single-agent learning algorithms rely on the Markov assumption but can deal with time-dependent non-stationarity [29].

In multi-agent learning systems, the process evolution is led by the agents' actions. For a learning agent, the choice of its future actions is dependent upon its own history, in other words upon past observations of its environment. If a past action of one agent have influenced the past evolution of the process, it could also have changed the learned behavior of a second agent. Then, from the first agent's perspective the future evolution of the environment can be different depending of its past action. Therefore, from a single agent's perspective, the environment no longer appears Markovian. This violation of basic assumptions requires new techniques to be developed to learn effective policies in stochastic games.

This paper contributes to analyze the underlying conditions that lead the environment to be non-Markovian from a single agent's perspective. We focus our study on independent learners in stochastic games where this problem is acute. Independent learners are non-communicative (isolated) agents that are unable to observe the rewards and actions of the other agents [8]. We formally show under which assumptions the local process is not Markovian. For this purpose, some new concepts are introduced like the divergent learning

*Corresponding author. E-mail: guillaume.laurent@ens2m.fr.

ISSN 1327-2314/11/$27.50

paths and the observability of the effects of others' actions. These findings are significant because they offer a new perspective on related research in multi-agent learning and help to provide a better understanding of the difficulties faced by multi-agent learning methods. It both helps to understand failures and successes of learning algorithms as well as being suggestive of areas for future work.

Beyond the formal analysis, we provide a case study example using independent Q-Learners. Q-Learning [37] is a well-known reinforcement learning algorithm that has been applied successfully in multi-agent cases despite the lack of guaranteed convergence. This example helps to thoroughly understand the impact of non-Markovian effects on this algorithm. Moreover, it shows that the exploratory actions can be detrimental for other learners in a cooperative game.

The remainder of this paper is organized as follows. Section II introduces the necessary background in multi-agent learning and related works. Section III defines formally the concept of learning agents. Section IV introduces some useful concepts and shows under which assumptions, the local process is not Markovian. Section V then gives a case study which enlightens the non-Markovian problem. Section VI concludes and explores some of the implications of the results described in the paper.

2. Background

In this section, we introduce some background material that will be used throughout the paper.

2.1. Stochastic games

Stochastic games [28] (also called Markov games) are the foundation for much of the research in multi-agent learning. Stochastic games are a superset of Markov decision processes [1] and matrix games [22, 23], including both multiple agents and multiple states.

Definition 1. A (finite) stochastic game is a tuple $\langle m, \mathcal{S}, \mathcal{A}^1, \ldots, \mathcal{A}^m, T, R^1, \ldots, R^m \rangle$, where,

- m is the number of agents (or players),
- $\mathcal{S}$ is the finite set of states,
- $\mathcal{A}^i$ is the finite set of actions available to agent i (and $\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \ldots \times \mathcal{A}^m$ is the joint action set),
- $T : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \mapsto [0; 1]$ is the transition function, such that,

$$\forall s \in \mathcal{S}, \forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \quad \sum_{s' \in \mathcal{S}} T(s, \boldsymbol{a}, s') = 1, \quad (1)$$

- $R^i : \mathcal{S} \times \boldsymbol{\mathcal{A}} \mapsto \Re$ is the reward function for the ith agent.

The transition function T gives the probability that action $\boldsymbol{a}$ in state s at time step k will lead to state s' at step $k+1$,

$$\mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{\boldsymbol{a}}_k = \boldsymbol{a}, \underline{s}_k = s] = T(s, \boldsymbol{a}, s'). \quad (2)$$

Note that joint actions and sets of joint actions are typeset in bold throughout the paper, e.g. $\boldsymbol{a}$ and $\boldsymbol{\mathcal{A}}$.

By definition, a stochastic game satisfies the Markov assumption. Basically, this requires that the probability to reach a state and to get a reward be fully specified by the current state and the agent's action (and eventually by the time). We can state this property formally.

Definition 2. A decision process is *Markovian*, iff, for all $k \in \mathbb{N}$, $(s_0, \ldots, s_{k+1}) \in \mathcal{S}^{k+2}$, and $(\boldsymbol{a}_0, \ldots, \boldsymbol{a}_k) \in \boldsymbol{\mathcal{A}}^{k+1}$,

$$\begin{aligned} \mathbb{P}[\underline{s}_{k+1} = s_{k+1} \mid \underline{\boldsymbol{a}}_k = \boldsymbol{a}_k, \underline{s}_k = s_k, \ldots, \\ \underline{\boldsymbol{a}}_0 = \boldsymbol{a}_0, \underline{s}_0 = s_0] \\ = \mathbb{P}[\underline{s}_{k+1} = s_{k+1} \mid \underline{\boldsymbol{a}}_k = \boldsymbol{a}_k, \underline{s}_k = s_k]. \end{aligned} \quad (3)$$

If a process is not Markovian, it is typically termed *history dependent*. Another quality of decision process is whether its transition and reward functions depend explicitly on time.time.

Definition 3. A decision process is *stationary* (or homogeneous), iff, for all $k, l \in \mathbb{N}$, $s', s \in \mathcal{S}$, $\boldsymbol{a} \in \mathcal{A}$

$$\begin{aligned} \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{\boldsymbol{a}}_k = \boldsymbol{a}, \underline{s}_k = s] = \\ \mathbb{P}[\underline{s}_{l+1} = s' \mid \underline{\boldsymbol{a}}_l = \boldsymbol{a}, \underline{s}_l = s]. \end{aligned} \quad (4)$$

A non-stationary process can be Markovian if the evolution of its transition and reward functions depends only on time step and not on the history of actions and states.

2.2. Independent learners

Claus and Boutilier [8] distinguish two fundamental classes of learning agents: independent learners and joint-action learners. The former have no knowledge of the other agents, interacting with the environment as if no other decision-makers exist. In particular, they are unable to observe the rewards and actions of the other agents. Independent learners were also described as isolated concurrent learners. Joint action learners, on the contrary, are aware of the existence of other agents

and are capable of perceiving (a posteriori) their actions and rewards.

Learning algorithms considering joint-action learners are easily implementable from standard single-agent reinforcement learning algorithms [16]. From a single agent's perspective the process stays Markovian as it observes the actions of others.

However, in many practical applications it is not reasonable to assume the observability of the actions of the others [20]. Most agents interact with their surroundings by relying on sensory information and action recognition is often far from trivial. With no knowledge of the actions of the other agents and payoffs, the problem becomes more difficult.

2.3. A key challenge

As an agent is not aware of actions of the other agents, it does not know if others are changing their policy or taking an exploratory action. Specifically, the transition probabilities associated with the action of a single agent from one state to another are not stationary and change over time as the other agents' action choices change. These choices are probably influenced by the past history of play, and so the history of play influences the future transition probabilities when revisiting a state. Therefore, from a single agent's perspective, the environment no longer appears Markovian. All the recent surveys and many works on multi-agent learning mention this fact as a key challenge [5,6,10,13,21,24, 26,32,35,38–40].

Using single-agent algorithm in this context is inherently flawed: the convergence of such algorithm usually relies on an underlying transition model that is stationary [35]. For example, straightforward extensions of Q-Learning to multi-agent systems fail to reach the optimal policy in fairly simple domains [8, 12]. However this approach has been successfully applied in simulation to domains such as prey-pursuing games [30], block-pushing problems [27], the control of elevators [9], the exploration of planets by mobile robot teams [41], the control of a two-link rigid manipulator [7], multi-robot cooperative transportation tasks [36], distributed manipulation [19] to name but a few.

Understanding the underlying conditions which cause non-Markovian aspects will be useful to predict whether a given system is likely to experience convergence problems or not. Moreover, once these conditions are isolated and understood, the development and application of new (or significantly modified versions of) machine learning methods will be greatly facilitated. Before examining these underlying conditions, we have to formalize the concept of a learner.

3. Definition of a Learner

In this section, we propose a formal definition of a learner. This definition is based on the concept of the individual policy used in most of the works.

3.1. Individual and Joint Policies

Independent learners observe the process state and receive a reward after executing an individual action. These agents choose their individual actions based on individual policies. A policy (or strategy) for an agent is a probability distribution over the set of its actions.

Definition 4. The policy of an agent i at time step k is a function $\pi_k^i : \mathcal{S} \times \mathcal{A}^i \mapsto [0;1]$ such that,

$$\forall s \in \mathcal{S}, \quad \sum_{a \in \mathcal{A}^i} \pi_k^i(s,a) = 1. \tag{5}$$

The policy of an agent gives the probability of selecting an action from a state,

$$\forall a \in \mathcal{A}^i, \quad \mathbb{P}[\underline{a}_k^i = a | \underline{s}_k = s] = \pi_k^i(s,a). \tag{6}$$

The space of possible policies for agent i is noted Π^i.

The concept of individual policy leads to the concept of joint policy when considering the actions of a group of agents.

Definition 5. The joint policy is a function $\boldsymbol{\pi}_k : \mathcal{S} \times \boldsymbol{\mathcal{A}} \mapsto [0;1]$ such that,

$$\forall s \in \mathcal{S}, \quad \sum_{a \in \mathcal{A}} \boldsymbol{\pi}_k(s,\boldsymbol{a}) = 1. \tag{7}$$

The joint policy is the policy of the group of agents. The joint policy is a mapping that defines the probability of selecting a joint action from a particular state, formally,

$$\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \quad \mathbb{P}[\underline{\boldsymbol{a}}_k = \boldsymbol{a} | \underline{s}_k = s] = \boldsymbol{\pi}_k(s,\boldsymbol{a}). \tag{8}$$

For independent learners which select actions independently, the joint policy is given by:

$$\forall s \in \mathcal{S}, \forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \quad \boldsymbol{\pi}_k(s,\boldsymbol{a}) = \prod_{i=1}^{m} \pi_k^i(s,\boldsymbol{a}^i). \tag{9}$$

3.2. Learners and divergent learning paths

A learner uses its past experiences to improve its behavior. The policy of a learner may change over time depending on the history of states, of rewards and of its own actions. It is clear that an agent may learn different behaviors depending on the evolution of the process. In other words, a learner is not only non stationary but also history dependent.

We propose here to formalize the definition of a learner. First, we call the learning path of an agent i at step k the random variable $\underline{h}_k^i$ which summarizes all past states, rewards and agent's actions,

$$\underline{h}_k^i = \left(\underline{s}_0, \underline{r}_0^i, \underline{a}_0^i, \underline{s}_1, \underline{r}_1^i, \underline{a}_1^i, \ldots, \underline{s}_k, \underline{r}_k^i, \underline{a}_k^i\right) \quad (10)$$

$\underline{h}_k^i$ takes on its values in $\mathcal{H}_k^i$ which is the set of possible realizations of the agent's learning path.

Definition 6. A learner i is defined by its *learning function* Ψ that associates a policy to a learning path,

$$\Psi : \begin{cases} \mathcal{H}_k^i \mapsto \Pi^i \\ h \ \rightarrow \Psi(h) = \pi_{k,h}^i \end{cases}, \quad (11)$$

where Π^i is the space of possible policies for agent i. $\pi_{k,h}^i$ is the policy of the agent i at step k after the learning path h.

This definition introduces only deterministic learning functions ($\forall h, g \in \mathcal{H}_k^i,\ h = g \Rightarrow \Psi(h) = \Psi(g)$). Stochastic learning functions could have been considered as well (see discussion bellow).

The second characteristic of a learner is that the obtained policy could be different with regard to the past of the agent even if the agent reaches the same state at step k. To express this, we propose the concept of divergent learning paths.

Definition 7. A pair of two different learning paths $h, g \in \mathcal{H}_k^i$ which lead to the same state s_k at time k, and such that,

$$\exists a \in \mathcal{A}^i, \quad \pi_{k,h}^i(s_k, a) \neq \pi_{k,g}^i(s_k, a). \quad (12)$$

is called a pair of *divergent learning paths*.

3.3. Discussion

The learning function represents the algorithm implemented in the agent. We focus here on deterministic learning functions. For instance, Q-Learning [37] is a deterministic learning function because agent's Q-values are updated using an algebraic equation and the agent's policy is usually computed from Q-values using a Boltzmann distribution [29].

Deterministic learning functions cover most of reinforcement learning algorithms for independent learners such as Wolf-PHC [4], Distributed Q-Learning [14], Hysteretic Q-Learning [17], etc.

We could also imagine stochastic learning functions. This case could be studied as well. The proof would be less straightforward but would lead to the same results. If deterministic learning functions imply non-Markovian local processes, so do stochastic ones. Indeed, focusing on deterministic functions is better for the clarity of the paper and does not harm the generality of the results.

4. Local decision processes

4.1. The Agent's perspective

Independent learners ignore the presence of the other agents in the system. That is to say, each agent can treat the other agents as part of its environment (cf. Figure 1). We call *local decision process* the environment from a single agent's perspective.

Definition 8. The local decision process from the ith agent's perspective is defined by the tuple $\langle \mathcal{S}, \mathcal{A}^i, \overline{T}_k^i, \overline{R}_k^i \rangle$, where,

- $\mathcal{S}$ is the finite set of states,
- $\mathcal{A}^i$ is the finite set of actions,
- $\overline{T}_k^i : \mathcal{S} \times \mathcal{A}^i \times \mathcal{S} \mapsto [0;1]$ is the transition function at time step k,
- $\overline{R}_k^i : \mathcal{S} \times \mathcal{A}^i \mapsto \Re$ is the expected reward at time step k.

The local transition function $\overline{T}_k^i$ and the local reward function $\overline{R}_k^i$ can be easily computed from T, R^i and from the current policies of the other agents [2,3].

Property 1. Let $\langle m, \mathcal{S}, \mathcal{A}^1, \ldots, \mathcal{A}^m, T, R^1, \ldots, R^m \rangle$ be a Stochastic Game with m agents and $\pi_k^i : \mathcal{S} \times \mathcal{A}^i \mapsto [0;1]$ the agent's policies. The local decision process from the ith agent's perspective is defined by the tuple $\langle \mathcal{S}, \mathcal{A}^i, \overline{T}_k^i, \overline{R}_k^i \rangle$, where for all $(s, a, s') \in \mathcal{S} \times \mathcal{A}^i \times \mathcal{S}$,

$$\overline{T}_k^i(s, a, s') = \sum_{a^{-i} \in \mathcal{A}^{-i}} \boldsymbol{\pi}_k^{-i}(s, \boldsymbol{a}^{-i}) T(s, \langle a, \boldsymbol{a}^{-i} \rangle, s'), \quad (13)$$

and,

$$\overline{R}_k^i(s, a) = \sum_{a^{-i} \in \mathcal{A}^{-i}} \boldsymbol{\pi}_k^{-i}(s, \boldsymbol{a}^{-i}) R^i(s, \langle a, \boldsymbol{a}^{-i} \rangle). \quad (14)$$

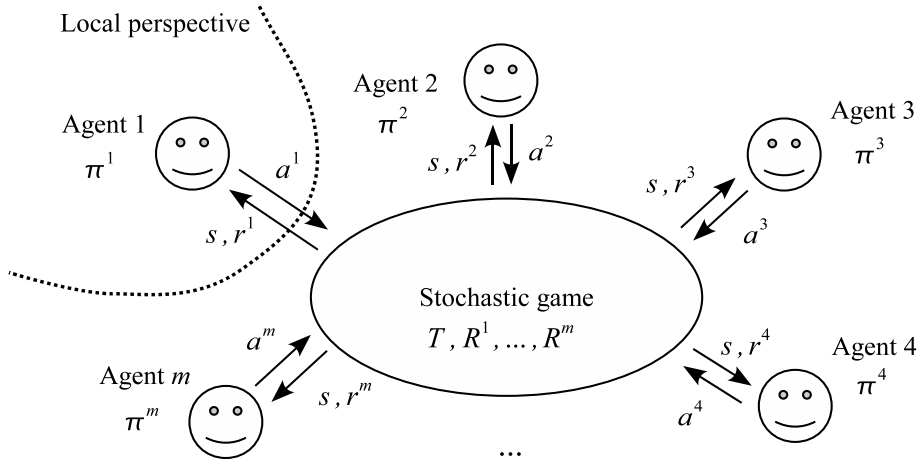


Fig. 1. Illustration of the first agent's perspective.

We use $\boldsymbol{\pi}_k^{-i}$ to refer to the current joint policy of all the agents except agent i. In the same way, we use $\boldsymbol{a}^{-i}$ to refer to the joint actions of all the agents except agent i. $\langle a, \boldsymbol{a}^{-i} \rangle$ is then the joint actions of all the agents including agent i.

Proof. This result comes directly from the law of total probability. For all $(s, a, s') \in \mathcal{S} \times \mathcal{A}^i \times \mathcal{S}$,

$$\begin{aligned}\overline{T}_k^i(s,a,s') &= \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{a}_k = a, \underline{s}_k = s] \\ &= \sum_{a^{-i} \in \mathcal{A}^{-i}} \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{a}_k = a, \\ &\quad \underline{\boldsymbol{a}}_k^{-i} = \boldsymbol{a}^{-i}, \underline{s}_k = s] \qquad (15) \\ &\quad \mathbb{P}[\underline{\boldsymbol{a}}_k^{-i} = \boldsymbol{a}^{-i} | \underline{a}_k = a, \underline{s}_k = s].\end{aligned}$$

As agents choose their action independently from each other, $\underline{\boldsymbol{a}}_k^{-i}$ and $\underline{a}$ are independent. So, we get,

$$\begin{aligned}\overline{T}_k^i(s,a,s') &= \sum_{a^{-i} \in \mathcal{A}^{-i}} \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{a}_k = a, \\ &\quad \underline{\boldsymbol{a}}_k^{-i} = \boldsymbol{a}^{-i}, \underline{s}_k = s] \qquad (16) \\ &\quad \mathbb{P}[\underline{\boldsymbol{a}}_k^{-i} = \boldsymbol{a}^{-i} | \underline{s}_k = s] \\ &= \sum_{a^{-i} \in \mathcal{A}^{-i}} T(s, \langle a, \boldsymbol{a}^{-i} \rangle, s') \boldsymbol{\pi}_k^{-i}(s, \boldsymbol{a}^{-i}).\end{aligned}$$

□

On the one hand, if the policies of other agents are stationary, the local process is simply reduced to a Markov decision process. On the second hand, it is obvious that the local process can be non-stationary as π_k^{-i} is not. We would like to show that, when one of the other agents is learning, the local process can no longer be Markovian.

4.2. Observability of the effects of the actions

In fact, everything depends on the observability of the effects of the actions of the learner. We say that the effects of the actions of an agent j are observable from the ith agent's perspective in a state s if the jth agent's choice has an influence on the transition probability of the process from the ith agent's perspective.

Definition 9. The effects of the actions of an agent j are observable from the ith agent's perspective in state s at step k if the probability of reaching the state s' from a state s depends on the jth agent's policy, formally if π^j and $\tilde{\pi}^j$ are two policies of agent j, such that,

$$\exists b \in \mathcal{A}^j,\ \pi^j(s,b) \neq \tilde{\pi}^j(s,b), \qquad (17)$$

Then, the effects of the actions of agent j are observable if,

$$\exists a \in \mathcal{A}^i, \exists s' \in \mathcal{S},\ \overline{T}^i_{\pi^j}(s,a,s') \neq \overline{T}^i_{\tilde{\pi}^j}(s,a,s'). \qquad (18)$$

This definition means that if the jth agent were to change its policy, the ith agent observe a change in the evolution of the process.

4.3. Non-markovian local decision processes

We are now able to state sufficient conditions of the local process to be non-Markovian.

Theorem 1. Let $\langle m, \mathcal{S}, \mathcal{A}^1, \ldots, \mathcal{A}^m, T, R^1, \ldots, R^m \rangle$ be a Stochastic Game with m agents and $\pi^i : \mathcal{S} \times \mathcal{A}^i \mapsto [0;1]$ the agent's policies. If,

– (h, g) are a pair of divergent learning paths for an agent j,

– h and g are both attainable,
– the effects of jth agent's actions are observable from an ith agent's perspective in the state s, where s is the common final state of h and g,

then, the local decision process from the ith agent's perspective is not Markovian.

A learning path is attainable if the probability to follow it is not zero.

Proof. Let $(h, g) \in \mathrm{H}_k^{j\,2}$ be a pair of divergent learning paths for an agent j. Let s be the common final state of h and g. As h and g verify the Eq. (12), we have,

$$\exists b \in \mathcal{A}^j, \quad \pi^j_{k,h}(s, b) \neq \pi^j_{k,g}(s, b). \tag{19}$$

As the effects of the actions of agent j are supposed to be observable, this statement implies that,

$$\exists a \in \mathcal{A}^i, \exists s' \in \mathcal{S}, \overline{T}^i_{\pi^j_{k,h}}(s, a, s') \neq \overline{T}^i_{\pi^j_{k,g}}(s, a, s'), \tag{20}$$

where s is the realization of $\underline{s}_k$.

$\overline{T}^i_{\pi^j_{k,h}}(s, a, s')$ is the transition probability of the local process after the path h, then we can write that,

$$\overline{T}^j_{\pi^j_{k,h}}(s_k, a, s') = \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{a}^i_k = a, \underline{s}_k = s_k, \underline{h}^j_k = h]. \tag{21}$$

So rewriting 20, we get,

$$\exists a \in \mathcal{A}^i, \exists s' \in \mathcal{S}, \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{a}^i_k = a, \underline{s}_k = s_k, \underline{h}^j_k = h] \neq \mathbb{P}[\underline{s}_{k+1} = s' \mid \underline{a}^i_k = a, \underline{s}_k = s_k, \underline{h}^j_k = g]. \tag{22}$$

Some transition probabilities depend on learning paths. As these learning paths are supposed to be attainable and different in the past, then the local process is history dependent and not Markovian. □

4.4. Discussion

The proof is straightforward as the definitions have been well established. What is important is the statement of conditions that lead to non-Markovian local processes.

The first condition expresses the fact that agent j is a learner. A learner is history dependent. But the presence of a learner is not sufficient to make the local process non-Markovian: some divergent learning paths of the learner must be attainable.

It is a strong assumption which is not easy to check in practice. Nevertheless, to guarantee their convergence, most of the reinforcement learning algorithms suppose that every state can be visited an infinite number of time. For instance, this is the case of Q-Learning. This condition is generally guaranteed by the use of action-selection methods which are not fully greedy and which keep a small account of random exploration. This practice permits the algorithm to visit the state space in a satisfactory way. This approach may also ensure that the learning paths of the learner are attainable.

The last hypothesis is that the effects of the actions of the learner must be observable from another agent's perspective. This new concept of observability reports a coupling of the actions of both agents. On the one hand, if there is no coupling between the actions then both agents will evolve independently in Markovian worlds. On the other hand, when actions are coupled, if the learner changes its policy, then the other agent will have to adapt itself to this evolution.

Anyway, these three assumptions are sufficient conditions. We believe that they are also necessary but a formal proof remains to be carried out. In the next section, we propose to examine an example. We think that a good example is as important as formal results to thoroughly understand non-Markovian impacts on learners.

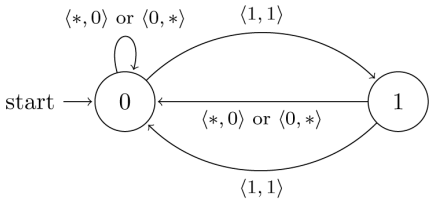
5. Case study

This example uses a two-state cooperative stochastic game with two Q-Learners. We chose Q-Learning [37] because many works use it with success despite the lack of guarantees (cf. Section 2.3). We will show that the local process from the first agent's perspective is not Markovian that explains the possible failures of using this algorithm even in cooperative multi-agent system. on learning processes.

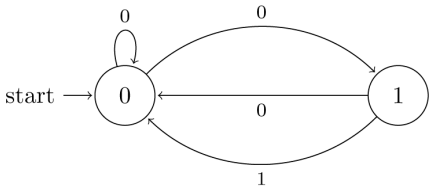
5.1. A simple stochastic game

We considered a stochastic game defined by:

– 2 states, $\mathcal{S} = \{0, 1\}$,
– 2 actions for each agent, $\boldsymbol{\mathcal{A}} = \{0, 1\} \times \{0, 1\}$,
– a deterministic transition function $T(s, \boldsymbol{a})$ and a team reward function R represented in Fig. 2.



(a) Transition function



(b) Team reward function

Fig. 2. Small stochastic game.

5.2. Q-Learners

Q-Learning is a well-known reinforcement learning algorithm for single agent process. The objective is to compute an optimal action-value function in order to get an optimal policy. The action-value function for a given policy π is defined by,

$$Q\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \underline{r}_{t+1} | \underline{s}_0 = s, \underline{a}_0 = a, \pi\right]. \quad (23)$$

where γ is a discount factor $0 \leqslant \gamma < 1$.

We assume that both agents are independent learners and apply Q-Learning in the classic sense. Each agent i stores a local action-value function $Q^i(s,a)$ only depending on state s and on its own action a. $Q^i(s,a)$ is an estimation of the expected sum of future rewards when taking action a^i from state s at time k. This approach results in big storage and computational savings in the action-space. For example, with 7 agents and 6 actions per agent only 42 Q-values have to be stored per state rather than 6^7 Q-values for joint-action learners.

The action-value function of agent 2 is noted $Q_k^2(s,a)$. We assume that Q_k^2 is equal to the optimal[1] action-value function at time k for the discounted criteria with $\gamma = 0.9$, i.e.,

$$Q_k^2(0,0) = 4.26 \qquad Q_k^2(1,0) = 4.26$$
$$Q_k^2(0,1) = 4.74 \qquad Q_k^2(1,1) = 5.26$$

We assume that agent 2 always follows a greedy policy based on Q_k^2 (no exploratory actions)

$$\pi_k^2(s,a) = \begin{cases} 1 \text{ if } a = \arg\max_b Q_k^2(s,b) \\ 0 \text{ else} \end{cases} \quad (24)$$

We also assume that agent 2 is still learning. We use the learning rate $\alpha = 0.6$ to quickly see non-Markovian effects. Agent 2 updates its Q-values using the one-step backup equation,

$$Q_{k+1}^2 = (1-\alpha) Q_k^2(s_k, a_k^2) + \alpha(r_{k+1} + \gamma \max_{b \in \mathcal{A}^2} Q_k^2(s_{k+1}, b)). \quad (25)$$

5.3. Two different scenarios

The local process from the first agent's perspective at step k is then defined by:

- 2 states, $\mathcal{S} = \{0,1\}$,
- 2 actions, $\mathcal{A}^1 = \{0,1\}$,
- a deterministic transition function, such that,

$$\overline{T}_k^1(s,a,s') = T(s, \langle a, \arg\max_b Q_k^2(s,b)\rangle, s') \quad (26)$$

- a reward function, such that,

$$\overline{R}_k^1(s,a) = R(s, \langle a, \arg\max_b Q_k^2(s,b)\rangle). \quad (27)$$

For example, the transition probability from the first agent's perspective starting in state 0, doing action 1 and reaching the state 1 is then,

$$\overline{T}_k^1(0,1,1) = T(0, \langle 1, \arg\max_b Q_k^2(0,b)\rangle, 1) = 1 \quad (28)$$

Now, we will study the effect of two different sequences of actions starting from state 0 at step k. In the first sequence, we assume that agent 1 follows the optimal policy[2], that is $\underline{a}_k^1 = 1$, $\underline{a}_{k+1}^1 = 1$. In the second one, agent 1 experiences other actions by playing $\underline{a}_k^1 = 0$, $\underline{a}_{k+1}^1 = 0$. Let us study in detail the effects of each scenario.

Scenario 1 (agent 1 is playing the optimal policy)

1. step k: the process state is 0, agent 1 does action 1 and agent 2 selects a greedy action in $\arg\max_b Q_k^2(0,b)$ that is action 1.

[1] The optimal action-value function is computed from a centralized point-of-view of the process.

[2] The optimal policy is computed from a centralized point-of-view of the process.

2. step $k+1$: the process reaches state 1, agent 2 updates its Q-values,

$$\begin{aligned} Q^2_{k+1}(0,1) &= (1-0.6)*4.74 + \\ &\quad 0.6*(0+0.9*5.26) \\ &= 4.74 = Q^2_k(0,1), \end{aligned} \tag{29}$$

then, agent 1 does action 1 and agent 2 chooses $\arg\max_b Q^2_{k+1}(1,b) = 1$

3. step $k+2$: the process returns to state 0, agent 2 updates its Q-values,

$$\begin{aligned} Q^2_{k+2}(1,1) &= (1-0.6)*5.26 + \\ &\quad 0.6*(1+0.9*4.74) \\ &= 5.26 = Q^2_k(1,1), \end{aligned} \tag{30}$$

The transition probability from the first agent's perspective for state 0 and action 1 is therefore:

$$\begin{aligned} &\overline{T}^1_{k+2}(0,1,1) = \\ &\quad T(0,\langle 1, \arg\max_b Q^2_{k+2}(0,b)\rangle, 1) = 1. \end{aligned} \tag{31}$$

So, following scenario 1 we get,

$$\begin{aligned} &\mathbb{P}[\underline{s}_{k+3} = 1 \mid \underline{a}^1_{k+2} = 1, \underline{s}_{k+2} = 0, \\ &\quad \underline{a}^1_{k+1} = 1, \underline{s}_{k+1} = 1,\ \underline{a}^1_k = 1, \\ &\quad \underline{s}_k = 0] = 1. \end{aligned} \tag{32}$$

Scenario 2 (agent 1 is exploring)

1. step k: the process state is 0, agent 1 does action 0 and agent 2 selects a greedy action in $\arg\max_b Q^2_k(0,b)$ that is action 1.
2. step $k+1$: the process state is 0 again, agent 2 updates its Q-values,

$$\begin{aligned} Q^2_{k+1}(0,1) &= (1-0.6)*4.74 \\ &\quad +0.6*(0+0.9*4.74) = 4.46, \end{aligned} \tag{33}$$

then, agent 1 does action 0 again and agent 2 chooses $\arg\max_b Q^2_{k+1}(0,b) = 1$

3. instant $k+2$: the process is still in state 0, agent 2 updates its Q-values,

$$\begin{aligned} Q^2_{k+2}(0,1) &= (1-0.6)*4.46 \\ &\quad +0.6*(0+0.9*4.46) = 4.19, \end{aligned} \tag{34}$$

The transition probability from the first agent's perspective for state 0 and action 1 is then:

$$\begin{aligned} &\overline{T}^1_{k+2}(0,1,1) = \\ &\quad T(0,\langle 1, \arg\max_b Q^2_{k+2}(0,b)\rangle, 1) = 0. \end{aligned} \tag{35}$$

So, following scenario 2 we get,

$$\begin{aligned} &\mathbb{P}[\underline{s}_{k+3} = 1 \mid \underline{a}^1_{k+2} = 1, \underline{s}_{k+2} = 0, \\ &\quad \underline{a}^1_{k+1} = 0, \underline{s}_{k+1} = 0, \underline{a}^1_k = 0, \underline{s}_k = 0] = 0. \end{aligned} \tag{36}$$

The transition probabilities of the local process have changed! This fact is not only a non-stationarity. Indeed, if the first agent's actions are different, the transition probabilities can remain the same. The transition probabilities of the local process for state 0 and action 1 depend on the past actions of agent 1. So, the local process is history dependent and not Markovian.

We can check that the action-state sequences form a pair of divergent learning paths for agent 2. The action-state sequences are different and lead to two different policies while reaching the same state 0 at time $k+2$. Both learning paths are attainable thanks to agent 1 exploratory actions.

We can also check that the effects of the actions of agent 2 are observable from the first agent's perspective in state 0. On the one hand, if $\pi^2(0,1) = 1$, then $\overline{T}^1_{k+2}(0,1,1) = 1$. On the other hand, if $\pi^2(0,1) = 0$, then $\overline{T}^1_{k+2}(0,1,1) = 0$. So, the Eq. (18) holds and the effects of the actions of agent 2 are observable.

5.4. Discussion

This example meets all the conditions for the local process to be non-Markovian: agent 2 is learning, the exploratory actions of agent 1 can induce divergent learning paths and the system is strongly coupled.

It is interesting to notice that if agent 1 acts in an optimal way (scenario 1), then agent 2 does not change. However, if agent 1 explores (scenario 2), then agent 2 destroys its initially-optimal policy. This highlights one of the dangers of the use of Q-Learning in the multi-agent context. Simple-but-coupled situations can put this algorithm in the wrong. This confirms and explains the failures of Q-Learning on matrix games presented by Claus and Boutillier [8] and by Kapetanakis and Kudenko [12].

6. Conclusion

In this paper, we contributed to identify conditions that lead the environment to be non-Markovian from an independent learner's perspective. The first assump-

tion requires that there is in the system at least one agent which is learning. This means that different state-action sequences can lead to different policies for this agent. The second condition is the *attainability* of such learning paths. The third condition is that the effects of the learner's actions are *observable* from another agent's perspective. These three assumptions are sufficient conditions. We believe that they are also necessary but a formal proof remains to be carried out. In the meantime, this analysis shows the proposed algorithms for independent learners in an interesting light.

Numerous works use directly single-agent algorithms in the multi-agent context. As we saw in the previous section, the results of this approach are not always good. For instance, the presented example shows that an exploratory action of an agent can destroy the optimal policy of another one. In this example the agents' actions are highly coupled. In low coupled distributed systems, it is reasonable to think that these algorithms are more likely to converge because the effects of the non-stationarity of agents are less *observable*. In some degree, the *observability* of the effects of the actions may have a direct influence on the convergence of the algorithms. Nevertheless, this comment must be qualified. Indeed, in cooperative multi-agent systems, theoretical and experimental works showed that only few couplings, as action shadowing, raise convergence issues [11,18]. Anyway, the convergence of these approaches requires to use decreasing exploration rates so as to avoid too much concurrent exploration [8,12]. The principle is to decrease the exploration frequency as the learning goes along so that each agent should find the best response to the behavior of the others. Another way is to use coordinated exploration techniques that exclude one or more actions from the agent's action space, so as to coordinate their exploration in a shrinking joint action space [33,34]. Both approaches can be considered as a means of reducing the *attainability* of divergent learning paths. By reducing the exploration of new actions, the agents evolve slower and the non-Markovian effects are reduced.

To overcome the problems encountered by single-agent methods, many algorithms have been proposed in order to address the non-stationarity problem. A few approaches address this by restoring the Markovian property of the local environment by the memorizing of past events [15,20]. The algorithm proposed by Lauer and Riedmiller [15] builds an exhaustive list of past actions [15]. Thanks to the distinction between joint actions, the algorithm is proved to converge. An obvious limitation is its application in large stochastic games because of the combinatorial explosion of the number of joint actions and so of the size of lists.

The last class of methods is to develop algorithms that are robust to non-Markovian effects. For instance, the Distributed Q-Learning ignores the effect of an unfortunate action done by a fellow agent [14]. The idea that the agents have to be lenient with regard to exceptional sub-optimal behavior of other agents gave several algorithms. We can mention the WOLF-PHC algorithm [4], the Hysteretic Q-Learning [17], and lenient learners [25]. The experimental results show that these approaches are especially robust and converge in most highly coupled systems with high observability of the effects of the actions. On the one hand, the leniency of agents tends to make the bad changes of policies of team-mates *unobservable*. On the other hand, an agent seldom changes its own policy, and then the number of *divergent learning paths* is lower than with single-agent methods.

In this paper, we limited ourselves to non-communicating learners because the non-Markovian effects are exacerbated in this case. The problem of non-stationarity agents also arises when communications are possible. Then, for cooperative systems, the communication between agents can be seen as a promising way to report policy changes and so to ensure convergence.

References

[1] R. Bellman, A markov decision process, *Journal of Mathematical Mechanics* **6** (1957), 679–684.

[2] V.S. Borkar, Reinforcement learning in markovian evolutionary games, *Advances in Complex Systems* **5**(1) (2002), 55–72.

[3] M. Bowling, Multiagent Learning in the Presence of Agents with Limitations, PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, May 2003.

[4] M. Bowling and M. Veloso, Multiagent learning using a variable learning rate, *Artificial Intelligence* **136** (2002), 215–250.

[5] M. Bowling and M.M. Veloso, An analysis of stochastic game theory for multiagent reinforcement learning, Technical Report CMU-CS-00-165, Computer Science Department, Carnegie Mellon University, 2000.

[6] L. Busoniu, R. Babuska and B. De Schutter, A comprehensive survey of multiagent reinforcement learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **3838**(2) (2008), 156–172.

[7] L. Busoniu, R. Babuska and B. De Schutter, *Decentralized Reinforcement Learning Control of a Robotic Manipulator*, In Proceedings of the 9th International Conference on Control, Automation,Robotics and Vision, pages 1347–1352, 2006.

[8] C. Claus and C. Boutilier, *The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems*, In Proc. of the National Conference on Artificial Intelligence, pages 746–752. AAAI, 1998.

[9] R.H. Crites and A.G. Barto, *Improving Elevator Performance using Reinforcement Learning*, In Proc. of Advances in Neural Information Processing Systems, Cambridge, MA, 1996. The MIT Press.

[10] J. Dowling and S. Haridi, Decentralized reinforcement learning for the online optimization of distributed systems, in: *Reinforcement Learning: Theory and Applications*, C. Weber, M. Elshaw and N.M. Mayer, eds, I-TECH Education and Publishing, 2008, pp. 143–166.

[11] N. Fulda and D. Ventura, *Predicting and Preventing Coordination Problems in Cooperative Q-Learning Systems*, In Proceedings of the International Joint Conference on Artificial Intelligence, 2007.

[12] S. Kapetanakis and D. Kudenko, *Improving on the Reinforcement Learning of Coordination in Cooperative Multi-Agent Systems*, In Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, Imperial College, London, April 2002.

[13] J.R. Kok and N. Vlassis, Collaborative multiagent reinforcement learning by payoff propagation, *Journal of Machine Learning Research* **7** (2006), 1789–1828.

[14] M. Lauer and M. Riedmiller, *An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems*, In Proc. of the Int. Conf. on Machine Learning, pages 535–542. Morgan Kaufmann, 2000.

[15] M. Lauer and M. Riedmiller, *Reinforcement Learning for Stochastic Cooperative Multi-Agent Systems*, In Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, pages 1516–1517, 2004.

[16] M.L. Littman, Value-function reinforcement learning in markov games, *Journal of Cognitive Systems Research* **2**(1) (2001), 55–66.

[17] L. Matignon, G.J. Laurent and N. Le Fort-Piat, *Hysteretic Q-Learning: An Algorithm for Decentralized Reinforcement Learning in Cooperative Multi-Agent Teams*, In Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, pages 64–69, San Diego, CA, USA, Oct 29-Nov 2 2007.

[18] L. Matignon, G.J. Laurent and N. Le Fort-Piat, Coordination of independent learners in cooperative markov games, Technical Report RR-2009-01, Institut FEMTO-ST/UFC-ENSMM-UTBMCNRS, Besançon, France, March 2009. http://hal.archives-ouvertes.fr/hal-00370889/fr/.

[19] L. Matignon, G.J. Laurent, N.L. Fort-Piat and Y.-A. Chapuis, Designing decentralized controllers for distributed-air-jet mems-based micromanipulators by reinforcement learning, *Journal of Intelligent and Robotic Systems* **59**(2) (2010), 145–166.

[20] F.S. Melo and M.C. Lopes, Convergence of independent adaptive learners. In Progress in Artificial Intelligence: 13th Portuguese Conf. on Artificial Intelligence, Lecture Notes in Artificial Intelligence, volume 4874, pages 555–567. Springer-Verlag, 2007.

[21] J.E. Munoz, A. Lazaric and M. Restelli, *Learning to Cooperate in Multi-Agent Social Dilemmas*, In Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, pages 783–785, 2006.

[22] J. Von Neumann and O. Morgenstern, Theory of Games and Economic Behavior. John Wiley and Sons, 1944.

[23] M.J. Osborne and A. Rubinstein, A Course in Game Theory. The MIT Press, 1994.

[24] L. Panait and S. Luke, Cooperative multi-agent learning: The state of the art, *Autonomous Agents and Multi-Agent Systems* **11**(3) (2005), 387–434.

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

[26] T. Sandholm, Perspectives on multiagent learning, *Artificial Intelligence* **171** (2007), 382–392.

[27] I. Sen, M. Sekaran and J. Hale, *Learning to Coordinate Without Sharing Information*, In Proc. of the National Conference on Artificial Intelligence, pages 426–431. AAAI, 1994.

[28] L.S. Shapley, Stochastic games, *PNAS* **39** (1953), 1095–1100, Reprinted in (Kuhn, 1997).

[29] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, 1998.

[30] M. Tan, *Multiagent Reinforcement Learning: Independent vs. Cooperative Agents*, In Proc. of the Int. Conf. on Machine Learning, pages 330–337, 1993.

[31] G. Tesauro, Reinforcement learning in autonomic computing: A manifesto and case studies, *IEEE Internet Computing* **11**(2) (2007), 22–30.

[32] K. Tuyls, P. Jan 'T Hoen and B. Vanschoenwinkel, An evolutionary dynamical analysis of multi-agent learning in iterated games, *Autonomous Agents and Multi-Agent Systems* **12** (2006), 115–153.

[33] K. Verbeeck, A. Nowé, J. Parent and K. Tuyls, Exploring selfish reinforcement learning in repeated games with stochastic rewards, *Proc of the Int Conf on Autonomous Agents and Multiagent Systems* **14**(3) (2007), 239–269.

[34] K. Verbeeck, A. Nowé, M. Peeters and K. Tuyls, Multi-agent reinforcement learning in stochastic single and multi-stage games. Lecture Notes in Computer Science, *Adaptive Agents and Multi-Agent Systems III* **3394** (2005), 275–294.

[35] N. Vlassis, *A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence*, In Ronald Brachman and Thomas Dietterich, editors, Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2007.

[36] Y. Wang and C.W. de Silva, A machine-learning approach to multi-robot coordination, *Engineering Applications of Artificial Intelligence* **21**(3) (2008), 470–484.

[37] C.J.C.H. Watkins and P. Dayan, Technical note: Q–learning, *Machine Learning* **8** (1992), 279–292.

[38] M. Weinberg and J. Rosenschein, *Best-Response Multiagent Learning in Non-Stationary Environments*, In Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, pages 506–513, 2004.

[39] E. Yang and D. Gu, Multiagent reinforcement learning for multirobot systems: A survey. Technical report, Department of Computer Science, University of Essex, 2004.

[40] H.P. Young, The possible and the impossible in multi-agent learning, *Artificial Intelligence* **171** (2007), 429–433.

[41] Z. Zheng, M. Shu-gen, C. Bing-gang, Z. Li-Ping and L. Bin, *Multiagent Reinforcement Learning for a Planetary Exploration Multirobot System*, In Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, pages 339–350, 2006.