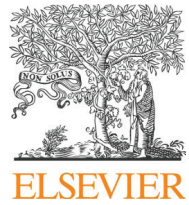

Procedia Computer Science

Volume 71, 2015, Pages 17–22

2015 Annual International Conference on Biologically Inspired Cognitive Architectures

Modeling Biological Agents Beyond the Reinforcement-Learning Paradigm

Olivier L. Georgeon, Rémi C. Casado, Laetitia A. Matignon

Université Lyon 1, LIRIS, UMR5205, F-69622, France.

Olivier.georgeon@liris.cnrs.fr, remi.casado@etu.univ-lyon1.fr, laetitia.matignon@univ-lyon1.fr

Abstract

It is widely acknowledged that biological beings (animals) are not Markov: modelers generally do not model them as agents receiving a complete representation of their environment's state in input (except perhaps in simple controlled tasks). In this paper, we claim that biological beings generally cannot recognize rewarding Markov states of their environment either. Therefore, we model them as agents trying to perform rewarding interactions with their environment (interaction-driven tasks), but not as agents trying to reach rewarding states (state-driven tasks). We review two interaction-driven tasks: the AB and AABB task, and implement a non-Markov Reinforcement-Learning (RL) algorithm based upon historical sequences and Q-learning. Results show that this RL algorithm takes significantly longer than a constructivist algorithm implemented previously by Georgeon, Ritter, & Haynes (2009). This is because the constructivist algorithm directly learns and repeats hierarchical sequences of interactions, whereas the RL algorithm spends time learning Q-values. Along with theoretical arguments, these results support the constructivist paradigm for modeling biological agents.

1 Introduction

The Reinforcement-Learning (RL) paradigm [e.g., 1] often appears as the most natural approach to designing unsupervised non-symbolic agents. Indeed, the RL conceptual framework was intended to account for the general problem of learning to achieve goals from experience of interaction, without predefined knowledge of the world, and without predefined semantics attached to perception and action.

In the RL paradigm, the agent's goals are synthetized in the form of a reward function related to the environment's states and transitions. RL techniques provide a sound theoretical basis for building agents that learn to maximize the discounted reward. At the core of the RL techniques is the elegant

Selection and peer-review under responsibility of the Scientific Programme Committee of BICA 2015

doi:10.1016/j.procs.2015.12.179

theory of Markov Decision Processes (MDPs). Formulating a given task as an MDP requires that the agent can observe the complete state of its environment. Unfortunately, this is not the case of natural agents (animals), making MDPs not suited to model natural agents. In this paper, we examine how Non-Markov Reinforcement Learning (NMRL) techniques can apply to modeling natural agents, and we investigate an alternative model called the embodied model.

2 The Reinforcement Learning paradigm

Sutton & Barto [1] proposed the RL formulation reported in Figure 1. The *Environment* rounded rectangle represents the real world or a simulated environment. The representation of the state s_t (equally called *state* or *state signal* by Sutton & Barto) is a data structure that represents the state of the environment on time *t*.

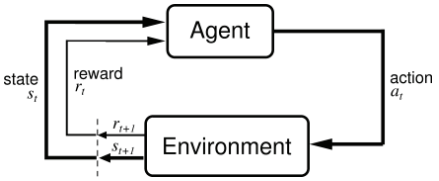

Figure 1: The agent-environment interaction in reinforcement learning [1, Figure 3.1]. On time *t*, the agent receives "*some representation* [s_t] *of the environment's state*" [1, §3.1]. The agent chooses an action a_t. The action a_t changes the environment. The agent receives a reward r_{t+1} from the resulting environment.

The dynamical system made of the agent and the environment is Markov if the next state signal s_{t+1} and the next reward r_{t+1} depend only on the previous state signal s_t and action a_t. This means that the state signal s_t constitutes a sufficient representation of the environment's state to allow the agent to choose the actions that lead to the reward. In this case, the environment can be represented by a Markov Decision Process (MDP) specified by a distribution of probability $T(s_{t+1}|s_t, a_t)$ that gives the probability to obtain any particular state s_{t+1} given s_t and a_t. The reward r_{t+1} can be implemented through a distribution of probability $\rho(r_{t+1}|s_t, a_t, s_{t+1})$, or, more simply, a scalar function $r(s_t, a_t, s_{t+1})$.

We found no arguments by RL theoreticians to justify the hypothesis that natural agents in the open real world can be modeled by MDPs. On the contrary, authors in psychology [e.g., 3] argue that perceiving the world consists of actively constructing a representation of the current situation through interaction, as opposed to directly receiving a representation of the world's state. A long time ago, some philosophers even argued that natural beings had no access to reality "as such" (*noumenal reality*, Kant), which we may interpret, in modern terminology, as having no access to the system's state, either considered internal or external to the agent. In this paper, we avoid the Markov hypothesis because we see no reason to believe that biological agents directly perceive their environment's state (except, perhaps, within some controlled experiments, e.g., monkeys pressing levers for reward).

3 Non-Markov Reinforcement Learning (NMRL)

The main approach to implementing reinforcement learning in non-Markov processes is based on the theory of Partially Observable MDPs (POMDPs). POMDPs are MDPs in which the state is not observable, but another "observation" signal stochastically related to the state is available to the agent. Figure 2 presents a typical formalization of a POMDP adapted from Spaan's article [2].

Yet, we are not satisfied with POMDPs for modeling natural agents in the real world because POMDPs use a Markov representation made by the designer a prior. In particular, if we don't want to model natural agents as if they had access to a Markov state signal, we should not either model them as if they had access to a reward associated with a Markov state. The same arguments against modeling natural agents by MDPs (presented in Section 2) also incite us to doubt that their goal can be

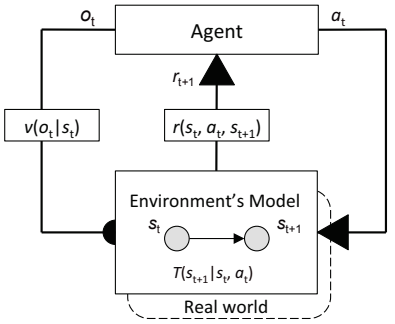

Figure 2: Typical formalization of a POMDP. The Environment's Model is an MDP that we can use to represent the real world. Similar to Figure 1, s_t denotes "some representation of the real world's state". Now, however, s_t is hidden to the agent. The agent's input data only consists of a partial observation o_t obtained through the distribution of probability $v(o_t|s_t)$. The agent is not Markov because o_{t+1} may depend on observations and actions anterior to t. The reward r_{t+1} is a function $r(s_t, a_t, s_{t+1})$ as it is in MDP models, rather than being directly provided by the real world as it is in Figure 1.

modeled through a reward function of a Markov state. As an analogy, if we agree that a robot has no sensor (internal or external) that would provide it with a Markov state signal, then we should also agree that the programmer of the robot couldn't program a reward function that includes a Markov state signal amongst its arguments. Therefore, we wish to avoid representing the agents' goals by a reward function that contains a Markov state amongst its arguments.

We acknowledge that POMDPs can be used to model agents whose goal is not directly related to the state of the environment. Some authors [e.g., 6] proposed tasks in which the agent seeks to perform rewarding behavioral patterns instead of reaching rewarding states. They do so by giving a decisive role to the a_t argument in the $r(s_t, a_t, s_{t+1})$ function. We call this kind of tasks interaction-driven tasks, in contrast with state-driven tasks in which the agent seeks to reach rewarding goal states. While POMDPs can be used to model interaction-driven tasks, we advocate an alternative model that eliminates the s_t and s_{t+1} arguments from the reward function. Section 4 presents this model. Sections 5 and 6 compare it with the POMDP model in example interaction-driven tasks.

4 The embodied model

The embodied model avoids the POMDP hypothesis that the environment is modeled as an MDP, and that the reward and the observation are functions of a Markov state. Not modeling the environment as a set of states implies removing the conceptual transition from state s_t to state s_{t+1}. It becomes then simpler to model the interaction cycle beginning with the action than with the observation, as illustrated by the black circle and triangle in Figure 3. Now, action a_t yields reward r_t rather than r_{t+1} (as is the case in Figures 1 and 2). We elaborate on this conceptual inversion of the interaction cycle, and on interaction-driven agents, in previous papers [5; 6].

Since the embodied model eliminates Markov states, it does not allow specifying the agent's goals by referring to the environment's states. Consequently, the embodied model only applies to interaction-driven tasks but not to state-driven tasks. This is acceptable for modeling natural agents, if we agree that natural agents do not seek to reach rewarding Markov states as we argued in Section 3.

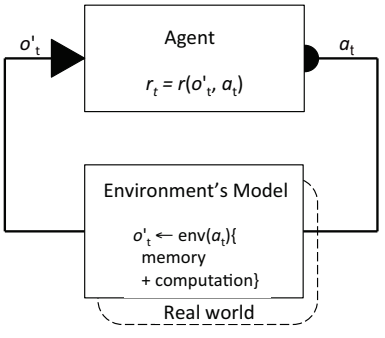

Figure 3: The embodied model. On time t, the agent chooses the action a_t (the black circle represents the conceptual beginning of the interaction cycle). The Environment's Model computes the observation o'_t through the program env(a_t) involving arbitrary memory and computation, making o' not Markov. We note the observation "o'" to highlight an important difference with POMDPs: o' is not a function of a hidden Markov state as o is in a POMDP. The resulting reward r_t is a function of the observation o'_t and possibly of the action a_t. (Technically, including a_t as an argument of r is not necessary since a_t can be passed through the environment as an attribute of o'_t, but we include it here for consistency with the POMDP model in Figure 2.)

5 Example interaction-driven tasks

Singh, Jaakkola, & Jordan [6] proposed a task here called the AB task. The agent must alternate between two actions to receive a positive reward. Figure 4 presents the POMDP model of the AB task.

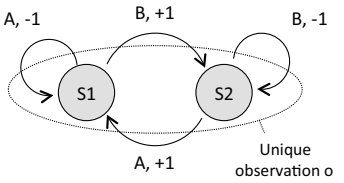

Figure 4: The AB task modeled as a two-state (S1 and S2) POMDP, adapted from [6]. The agent has two actions A and B; one will deterministically swap the environment to the other state, whereas the other action has no effect on the state. If the agent jumps to the other state, it receives reward +1, otherwise -1. Both states generate the same observation *o*, which makes *o* uninformative. The optimal policy consists of alternating action A and B.

The AB environment can be implemented according to the embodied model through the program $o'_t \leftarrow \text{env}(a_t)$ {if $a_{t-1} \neq a_t$ then $o'_t \leftarrow o'^1$ else $o'_t \leftarrow o'^2$}; $r(o'^1) = 1$, $r(o'^2) = -1$. Note that **observations o'^1 and o'^2 do NOT constitute partial representations of the state** of the hidden MDP: each state S1 and S2 would give rise to either o'^1 or o'^2 depending on the action, making o'_t alone uninformative about the state, just as *o* in the POMDP proposed by Singh et al. (Figure 4).

Georgeon, Ritter & Haynes [7] proposed an extension of the AB task, here called the AABB task. The agent must repeat the same action only twice in a row to receive a positive reward; otherwise it receives a negative reward. The AABB task could be modeled as a 4-state, 1-observation POMDP, but we implemented it within the embodied model through the program $o'_t \leftarrow \text{env}(a_t)$ {if $a_{t-2} \neq a_t$ and $a_{t-1} = a_t$ then $o'_t \leftarrow o'^1$ else $o'_t \leftarrow o'^2$}; $r(o'^1) = 1$, $r(o'^2) = -1$.

6 Non-Markov Reinforcement Learning (NMRL) experiment

In a POMDP, since the agent's observations are not sufficient to uniquely identify the state, a direct mapping of observations to actions (i.e., memory-less policy) will not achieve an optimal behavior. For instance, in the AB task, there are two memory-less deterministic policies: always execute A and always execute B. The best memory-less stochastic policy will choose either action 50% of the time. None of these memory-less policies are optimal. For an agent to consider optimal decision in a POMDP, memory is needed. Incorporating some form of memory in the agent was among the first solutions to deal with NMRL agents [2]. The idea is to construct a Markov or nearly-Markov signal from historical sequences composed of observations received and actions taken by the agent. It is then possible to use classical RL algorithms, such as Q-learning [e.g., 8], to compute an optimal deterministic mapping of historical sequences to actions.

If the considered POMDP has a relatively small order (i.e. the knowledge of an history of length 2 or 3 is sufficient to predict the evolution of the system), then it will be possible to construct a Markov signal based on memory. But with higher orders, or large observation sets, the algorithm suffers from the growth of the state space. Some algorithms [e.g., 9; 10] are searching historical sequences with minimal length to construct memory-based observations for the optimal decision. These historical sequences are composed of non-ambiguous elements.

Dutech [9] called the historical sequences Observation-Action Trajectories (OATs). An OAT of order *n* is a series of *n* consecutive observations and actions plus a final observation: $o_1.a_1 \ldots o_n.a_n.o_{n+1}.$, with $o_1 \ldots o_{n+1} \in O$ and $a_1 \ldots a_n \in A$. In the POMDP representations of the AB and AABB tasks, the set of observations is $O = \{o\}$, and the set of actions is $A = \{A, B\}$. The algorithm performs Q-learning using the set of current OATs as the set of states, increasing *n* until the agent manages to master the task satisfactorily. To increase *n*, only the most used or ambiguous OATs are extended; ambiguous OATs are those whose Q-values are close and converge slowly [9; 10].

Our implementation of this algorithm works as well when we model these tasks according to the embodied model. In this case, however, it is more logical to construct Action-Observation Trajectories (AOTs) than OATs because of the conceptual inversion of the interaction cycle. An AOT of order n is a sequence $a_1.o'_1 \ldots a_n.o'_n$, with $a_1 \ldots a_n \in A$ and $o'_1 \ldots o'_n \in O'$; $A = \{A, B\}$, $O' = \{o'^1, o'^2\}$. Note again that using o' instead of o makes little difference because the next choice is determined by the previous actions.

In our experiments, the agent mastered the AB task with AOTs of order $n = 1$, and the AABB task when enough AOTs had reached $n = 2$. This is because the AOTs must cover the task's temporal dependency to allow the agent to make the right decision. Figure 5 plots the results obtained in the AABB task. Table 1 shows the Q-values obtained on step 840 when the agent had learned the task. These results are consistent with Dutech's [9]. Due to lack of space, the results using OATs in the POMDP model are not reported in this article but are similar.

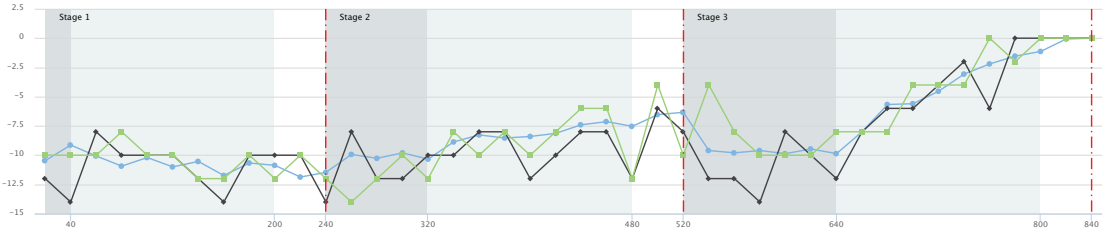

Figure 5: Results of the AOT algorithm applied to the AABB task: average over 30 runs (blue), best run (green), and worst run (black). X-axis: steps 0 to 840. Y-axis: total reward over the last 20 steps. Each stage begins with a phase of detection of the most ambiguous AOTs (dark grey 40 × Stage # steps, uniform Q-learning, $\alpha = 0.1$, $\gamma = 0.9$). Next 160 steps (light grey): learns the Q-values for these AOTs ($\varepsilon = 0.1$). Next 40 steps (white): full greedy. After each stage, the three most ambiguous AOTs are extended ($n \leftarrow n + 1$, beginning with $n = 1$). Stage 3: the agent masters the AABB task from step 820 on, receiving the best average reward possible (0), when enough AOTs have reached order $n = 2$.

Table 1: Most significant AOTs (columns) and their Q-values for each action (lines) learned on step 840. Q-values are the discounted expected reward for choosing a given action. The full greedy policy consists of choosing the action that has the greatest Q-value in the context of a particular AOT.

	Ao2Ao2	Bo1Ao2	Bo2Ao2	Ao1Bo2	Ao2Bo2	Bo1Bo2	Bo2Bo2	Bo2Bo1	Ao1Ao2	Ao2Ao1
A	-0.23	0.51	0.44	-0.13	-0.08	-0.27	-0.21	-0.41	-0.31	-0.47
B	-0.22	-0.10	-0.09	0.50	0.44	-0.29	-0.20	-0.45	-0.29	-0.41

7 Conclusion

We have shown that both the POMDP model and the embodied model can be used to represent the AB and the AABB tasks. This comparison is intended to help readers—specifically from the RL community—better understand the similarities and differences between the two models. Results show that both models work using historical sequence RL algorithms.

The POMDP model and the embodied model differ conceptually by the fact that the embodied model materializes two commitments: a) the reward function does not include a Markov state amongst its arguments, and b) the observation o' does not constitute a partial representation of the environment's state. These commitments allow easily transferring the agent's algorithm to a robot whose sensors do not return a Markov state signal. However, they imply that the robot's goals consist of performing rewarding behaviors rather than reaching rewarding states.

In 2009, Georgeon, Ritter, & Haynes [7] studied the AB and AABB tasks following a constructivist approach based upon the embodied model. They proposed an algorithm that can learn

the AABB task in less than 20 interaction cycles [7, Figure 1]. This makes a striking difference with the 820 cycles needed by the historical sequence RL algorithm reported in Figure 5. This difference comes from the fact that the constructivist algorithm [7] directly focuses on recording and repeating sequences of interactions, while the historical sequence RL algorithm needs time learning Q-values.

Together with Georgeon, Ritter, & Haynes's paper, this paper allows comparing two paradigms of agent modeling: the RL paradigm (in this paper, using the POMDP model or the embodied model) and the constructivist paradigm ([7], using the embodied model). We conclude that both paradigms can be used to model biological agents. The chosen paradigm, however, may impact how the designer designs the algorithms. The philosophy of the RL paradigm consists of transforming a non-Markov task into a Markov task by constructing states based upon sequences of interactions (OATs or AOTs), and then applying Markov techniques (e.g., Q-learning). The philosophy of the constructivist paradigm consists of recording hierarchical sequences of interactions (inspired by Piaget's sensorimotor schemes [11]) and reusing these sequences directly.

While the two paradigms are possible, the present study incites us to support the constructivist paradigm and the embodied model for modeling biological agents because of the following reasons: a) the constructivist paradigm proved more efficient in learning the AABB task; b) we find the embodied model more elegant because it does not require referring to Markov states and to the next interaction cycle (t+1); and c) the constructivist paradigm and the embodied model better comply with cognitive theories [e.g., 11], constructivist epistemology, and philosophy.

Future studies may investigate other NMRL techniques (e.g., actor-critic) in other interaction-driven tasks (e.g., stochastic tasks), but we expect that they would not significantly contradict our conclusions as long as these techniques are based on a reward function that includes a Markov state, and on observations that are partial representations of the environment's states.

References

[1] Sutton, R. & Barto, A. (1998). Reinforcement learning: An introduction. Cambridge, MA: MIT Press.

[2] Spaan M. (2012). Partially Observable Markov Decision Process. In "Reinforcement Learning: State of the Art". M. A. Wieing & M. van Otterlo Eds. Springer Verlag.

[3] Findlay, J., & Gilchrist, I. (2003). Active Vision: The Psychology of Looking and Seeing. USA: Oxford University Press.

[4] Georgeon, O. & Cordier, A. (2014). Inverting the interaction cycle to model embodied agents. Procedia Computer Science, 41, pp 243-248. The Fifth international conference on Biologically Inspired Cognitive Architectures. Boston, MA.

[5] Georgeon, O., Marshall J., & Gay S. (2012). Interactional motivation in artificial systems: between extrinsic and intrinsic motivation. Second International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EPIROB 2012). San Diego, CA. 1-2.

[6] Singh S., Jaakkola T., Jordan M. (1994). Learning without state-estimation in partially observable Markovian decision processes. International Conference on Machine Learning.

[7] Georgeon O., Ritter F., & Haynes S. (2009). Modeling Bottom-Up Learning from Activity in Soar. 18th Annual Conference on Behavior Representation in Modeling and Simulation (BRIMS 2009). Sundance, Utah. 65-72.

[8] Watkins C. and Dayan P. (1992) "Technical note: Q-learning," Machine Learning, vol. 8, pp. 279–292.

[9] Dutech A. (2000). Solving POMDPs using selected past events. ECAI, pp. 281-285.

[10] McCallum A. (1996) Reinforcement learning with selective perception and hidden state. PhD thesis, University of Rochester

[11] Piaget, J. (1951). The psychology of intelligence. London: Routledge and Kegan Paul.