Reinforcement Symbolic Learning

Chloé Mercier, Frédéric Alexandre, Thierry Viéville

Reinforcement Symbolic Learning *

Chloé Mercier[0000−0002−5069−3138], Frédéric Alexandre[0000−0002−6113−1878], and Thierry Viéville[0000−0003−1031−3572]

Mnemosyne Team, Inria Bordeaux, LaBRI and IMN

Abstract. Complex problem solving involves representing structured knowledge, reasoning and learning, all at once. In this prospective study, we make explicit how a reinforcement learning paradigm can be applied to a symbolic representation of a concrete problem-solving task, modeled here by an ontology. This preliminary paper is only a set of ideas while feasibility verification is still a perspective of this work.

Keywords: Reinforcement Symbolic Learning· Ontology Edit Distances· Models for Learning Sciences

1 Introduction

Understanding how humans solve problems and learn is a key issue in education, and one of the transversal competencies sometimes referred to as "21st-century skills". At the cognitive level, we need to consider both exploration and exploitation strategies directed either by a stimulus-driven behavior or toward different concurrent goals in a goal-driven behavior [1]. At a computational level, the typical paradigm is reinforcement learning (RL) with a final reward (success of the task) and some intermediate rewards (discoveries of affordances, partial result regarding the goal), more precisely intrinsically motivated reinforcement learning [9]. The latter family of models is extensively used in cognitive neurosciences to model high-level executive control functions (e.g., [2,8]).

In a recent development, we have introduced the operationalization of a creative problem-solving task, via the construction of an ontology[1] [6]. The state is defined by the configuration of the objects manipulated during the task and some other observables regarding the learner[2]. The key point is that, in our case, the internal state of the subject and the external state of the task material constitute a complex structure, modeled here by a set of "statements"; that is to say, in the ontological vocabulary, entities typed by classes and linked together

* Supported by Inria, AEx AIDE https://team.inria.fr/mnemosyne/en/aide.

[1] Concretely the ontology role is to help specifying the representation of the world as conceived by the learner, i.e., what is observed and what is to be inferred during the learning activity, to solve the task. This also helps to verify the specification coherence of the model, which allows us to infer some assumptions about non observable elements of the learning process, as detailed in the supplementary material accessible here https://gitlab.inria.fr/line/aide-group/creacog.

[2] See here for a video illustration.

through relationships labeled by properties. Unlike usual mechanisms based on Markov chains, the state space to consider in our RL setup is thus not reduced to an unordered finite enumeration. We would like to study here to what extent reinforcement learning could be designed on such state spaces.

In "symbolic reinforcement learning" (e.g. [3]), deep neural networks transform raw perceptual data into a symbolic representation which is then fed to a symbolic module that might perform, for instance, action selection. Other approaches such as [4] propose architectures where a numerical reinforcement algorithm communicates with a reasoner. Here we would like to explore another track and make the reinforcement algorithm work directly on the symbolic data space itself in a more integrative way, which we propose to call "reinforcement symbolic learning".

2 Symbolic state space specification

General framework: An agent interacts with its environment. At a given discrete time, it perceives a part of the environment, i.e., a stimulus, including a reward. It infers elements (e.g., causes) from this input cue, including the computation of the next action. In our case, we consider a potentially hypermnesic agent for which any previous input, including rewards, might be part of the internal state. This choice is directly linked to the notion of episodic memory, an episode here being represented by an event sequence (a list of times of occurrence of atomic events). This frees us from the "Markovian" constraint taking into account not just one step in the past, at the cost of a multiplicative increase of the state space. This is going to be manageable thanks to the hierarchical structure of our state space, which encompasses a lot of information without the need for an exhaustive enumeration. Furthermore, the state space no more reduces to a flat enumeration of state values, while the number of possible ontology construction of size S on a vocabulary if size V is an order of magnitude higher (i.e. roughly in $O(V^3\,S)$).

Input structure: The input is a hierarchical data structure time sequence, with information regarding the task and the learner. Syntactically this corresponds to tuples of named values $\{\cdots, \texttt{name} : \texttt{value}, \cdots\}$; each value has a "type" (determined by a predefined schema [3] making explicit the value set), and may be either another tuple or a literal or the meta-value `undefined` (i.e., expected by the schema, but not present in the given context). Literal values are taken among a finite enumeration of qualitative values (e.g., a color set) or quantitative values (i.e., finite precision bounded values).

Comparing two inputs: An input s is thus a tree data structure, equipped[4] with a *a partial semi-order* compatible with an *extended semi-distance*. This means

[3] In the sense of https://json-schema.org.

[4] We consider *edit operations* given an input (l+) adding, (l-) deleting or (l#) changing a value in a list, (t+) defining, (t-) undefining or (t#) changing a value in a tuple,

that two values may be equal, indistinguishable (i.e., too close to be ordered, thus equal or not), comparable or incomparable (i.e., too different to be compared). This mechanism not only allows to define a distance between two inputs (as the minimal cost of editing sequences transforming one input into another), but also to make explicit which node has been added, deleted, or changed. Thus, it offers a "geodesic", i.e., a path of transformation from one structure to another, allowing us to interpolate intermediate input structure between both of them.

Inferring other elements from input: Each data structure is translated in terms of RDF statements as follows. Each tuple is a "subject" and each named value corresponds to a relationship labeled by a "property", the value being the "object" targeted by the relationship. This transformation allows us to generate an ontology[5], offering the possibility to perform inferences, thus implementing the learner behavior at a pure symbolic level. Conversely, each RDF ontology graph may be mapped back onto the data structure, with tuples having the value `undefined` if the corresponding statement is absent from the ontology after reasoning, and defined as the property object·s otherwise — hence the need for a pre-defined schema.

3 Reinforcement learning on symbolic state space.

Let us consider a concrete example on the most common algorithm setup, i.e., Q-learning with ϵ-greedy exploration, namely an action-state value function $Q : S \times A \to \mathbb{R}$. At each step this function is updated, using the weighted average of the old value and the new information, while the action is chosen to either maximize the reward value, given the state, or with a small probability randomly explore new actions. This algorithm is known as "model-free", however, we are using it in a non-conventional way with inferences we generate from prior information on symbolic states, bringing it somewhat closer to model-based algorithms.

The key point is that, given the largeness of the state space, each state value is very likely different from another, so that one state value is very likely visited once, making it impossible to use the usual update rule on tabulated values $Q[s_t, a_t]$. However, given a new state value s_t and reward r_{t+1}, we can easily update all preceding Q-values in a neighborhood.

Considering an exponential weighting of radius ρ, for a learning factor α, a discount factor γ, this writes, for all Q-values:

$$Q[s, a_t] \mathrel{+}= \alpha\, e^{-d(s,s_t)/\rho}\ (r_{t+1} + \gamma\ \max_a Q(s_{t+1}, a) - Q[s, a_t])$$

where $d(s, s_t)$ stands for the predefined edit distance between both states. During an epoch of T steps, it means that we have to compute $O(T^2)$ edit distances,

each of these operations having a user-defined positive cost, related to the literal extended semi-distances. The key point is that we consider restricted edit distances preserving the tree filiation, computable in polynomial time [7], which would not have been the case otherwise, or if considering the tree as a general graph or ontology portion.

[5] In the sense of RDF and OWL2.

and Q-value updates, but this complexity depends only on the trajectory length, rather than on the state space itself.

The computation of the maximum reward prediction $\max_a Q(s_{t+1}, a)$, requires to both (i) interpolate the Q-value given available tabulated values and (ii) enumerate action candidates, including unprecedented actions. For (ii) consider a set of potential actions a_k including previous actions, predefined prototypical actions, and putative actions generated by an external process. For (i) we can use an exponential interpolation

$$Q(s, a_k) = \sum_{s_t, a_t} e^{-(d(s,s_t)+d(a_k,a_t))/\rho}\, Q[s_t, a_t] \Big/ \sum_{s_t, a_t} e^{-(d(s,s_t)+d(a_k+,a_t))/\rho}$$

in coherence with the previous design choice.

4 Conclusion

We have here all ingredients to apply well-established Q-learning mechanisms not only on an enumeration of indexed states, but on a rich semantic structure, which was the goal of this preliminary work. Meanwhile, in a companion study, we explore ways to map such a semantic structure onto a neuronal vector space [5]. We are making a prospective presentation here to share the scientific idea, it goes without saying that this is only an open issue to be developed.

References

1. Alexandre, F.: A global framework for a systemic view of brain modeling. Brain Informatics **8**(1), 3 (Dec 2021)
2. Domenech, P., Koechlin, E.: Executive control and decision-making in the prefrontal cortex. Current Opinion in Behavioral Sciences **1**, 101–106 (Feb 2015)
3. Garnelo, M., Arulkumaran, K., Shanahan, M.: Towards Deep Symbolic Reinforcement Learning. arXiv:1609.05518 [cs] (Oct 2016), arXiv: 1609.05518
4. Ma, Z., Zhuang, Y., Weng, P., Li, D., Shao, K., Liu, W., Zhuo, H.H., Hao, J.: Interpretable Reinforcement Learning With Neural Symbolic Logic. ICLR 2021: 9th International Conference on Learning Representations (Sep 2020)
5. Mercier, C., Chateau-Laurent, H., Alexandre, F., Viéville, T.: Ontology as neuronal-space manifold: towards symbolic and numerical artificial embedding. In: KRHCAI 2021 Workshop @ KR 2021: 18th International Conference on Principles of Knowledge Representation and Reasoning. Hanoi, Vietnam and/or Online (Nov 2021), submitted
6. Mercier, C., Roux, L., Romero, M., Alexandre, F., Viéville, T.: Formalizing Problem-Solving in Computational Thinking : an Ontology approach. In: ICDL 2021: IEEE International Conference on Development and Learning. Beijing, China (Aug 2021)
7. Ouangraoua, A., Ferraro, P.: A Constrained Edit Distance Algorithm Between Semi-ordered Trees. Theoretical Computer Science **410**(8-10), 837–846 (2009), publisher: Elsevier
8. Rmus, M., McDougle, S.D., Collins, A.G.: The role of executive function in shaping reinforcement learning. Current Opinion in Behavioral Sciences **38**, 66–73 (Apr 2021)
9. Singh, S., Lewis, R.L., Barto, A.G., Sorg, J.: Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective. IEEE Transactions on Autonomous Mental Development **2**(2), 70–82 (Jun 2010)