Early Empirical Results on Reinforcement Symbolic Learning

Waris Radji, Corentin Léger, Lucas Bardisbanian

Early Empirical Results on Reinforcement Symbolic Learning

Waris Radji, Corentin Léger, Lucas Bardisbanian

RESEARCH REPORT
N° 9509
May 2023
Project-Teams Mnemosyne

ISSN 0249-6399 ISRN INRIA/RR--9509--FR+ENG

Early Empirical Results on Reinforcement Symbolic Learning

Waris Radji*, Corentin Léger†*, Lucas Bardisbanian†*

Project-Teams Mnemosyne

Research Report n° 9509 — May 2023 — 9 pages

Abstract: Reinforcement learning is a subfield of machine learning that is concerned with how agents learn to make decisions in an environment in order to maximize some notion of reward. It has shown a great promise in a variety of domains, but it often struggles with generalization and interpretability due to its reliance on black-box models. One approach to address this issue is to incorporate knowledge representation into reinforcement learning. In this work, we explore the benefits of using symbolic representation in reinforcement learning and present preliminary results on its performance compared to standard reinforcement learning techniques. Our experiments show that the use of symbolic representation can significantly improve the generalization capabilities of reinforcement learning agents. We also discuss the potential challenges and limitations of this approach and suggest avenues for future research.

Key-words: Reinforcement Symbolic Learning, Ontology, Edit Distances, Learning Sciences.

Work done during a Master project at ENSEIRB-MATMECA.

* Bordeaux Institute of Technology - ENSEIRB-MATMECA
† Bordeaux Institute of Technology - ENSC

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Premiers résultats empiriques sur l'apprentissage par renforcement symbolique

Résumé : L'apprentissage par renforcement est un sous-domaine de l'apprentissage automatique qui s'intéresse à la manière dont les agents apprennent à prendre des décisions dans un environnement afin de maximiser une récompense. Il s'est avéré très prometteur dans divers domaines, mais il se heurte souvent à des problèmes de généralisation et d'interprétabilité en raison de sa dépendance à l'égard de modèles de type "boîte noire". Une approche pour résoudre ce problème consiste à incorporer la représentation des connaissances dans l'apprentissage par renforcement. Dans ce travail, nous explorons les avantages de l'utilisation de la représentation symbolique dans l'apprentissage par renforcement et présentons des résultats préliminaires sur sa performance par rapport aux techniques d'apprentissage par renforcement standard. Nos expériences montrent que l'utilisation de la représentation symbolique peut améliorer de manière significative les capacités de généralisation des agents d'apprentissage par renforcement. Nous discutons également des défis potentiels et des limites de cette approche et suggérons des pistes de recherche pour l'avenir.

Mots-clés : Apprentissage symbolique par renforcement, Distances d'édition, Ontologie, Sciences de l'éducation.

1 Introduction

The human brain has evolved to process and interpret the world symbolically [3], by utilizing abstract concepts such as language and reasoning, and by drawing connections between seemingly dissimilar observations, such as perceiving oranges and lemons as similar due to their shared sensory characteristics. However, classical artificial intelligence agents operate differently by processing information through numerical values and statistical patterns, which may not always be intuitively understandable to humans.

One popular approach to reinforcement learning is Q-learning [4], which uses a value function known as the Q-function to estimate the expected cumulative reward for each action in a given state.

The Q-function is updated iteratively using the Bellman equation, which expresses the expected value of the Q-function in terms of the expected reward for taking a given action in a given state, plus the discounted expected value of the Q-function in the resulting state. The Bellman equation is expressed as follows:

$$Q[s, a_t] += \alpha(r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q[s, a_t]) \quad (1)$$

Here, s_t and a_t represent the current state and action, r_{t+1} represents the immediate reward received after taking action a_t in state s_t, γ is the discount factor (used to give less weight to future rewards), and α is the learning rate (used to control the rate at which the Q-function is updated).

Q-learning has proven to be a powerful technique for a variety of problems, including game playing, robotics, and control systems. However, one limitation of Q-learning is that it can struggle to generalize to new and unseen states. This is because Q-learning relies on an explicit representation of the state space, which can be difficult to achieve for complex and high-dimensional environments.

In order to address this limitation and better understand how humans learn, the Inria Mnemosyme team has introduced a paper where they present an alternative of Q-Learning with symbolic data, called *reinforcement symbolic learning* [2]. This technique is based on the fact that it is possible to define an edit distance between two symbolic objects under certain representation constraints, and therefore use the knowledge acquired from previous experiences to adapt faster and better with new states.

In the context of symbolic objects, the edit distance is a metric that measures the minimum number of operations (insertions, deletions, or substitutions) required to transform one symbolic object into another. It is a way of quantifying the similarity between two symbolic objects.

Considering a space of symbolic states S in a reinforcement learning environment, a distance d is defined as $\forall s_i \in S, s_j \in S, d(s_i, s_j)$. The operation of updating the Q-value (at each step in the environment), proposed by the *reinforcement symbolic learning* approach is :

$$Q[s, a_t] += \alpha e^{-d(s,s_t)/\rho}(r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q[s, a_t]) \quad (2)$$

where ρ is the exponential weighting of radius, α the learning factor and γ the discount factor. In classical Q-Learning, a Q-Value is updated at each iteration. In this approach all Q-Values are updated according to the radius ρ which updates with a stronger intensity the states s which are closest to s_t according to the edit distance. This theoretically allows to have agents that have better generalization capabilities, which can also be similar to embedding generation in deep reinforcement learning.

In this paper we present some preliminary results of the application of reinforcement symbolic learning on a self-defined environment and edit distance. Some details of the implementation of *reinforcement symbolic learning* will also be given.

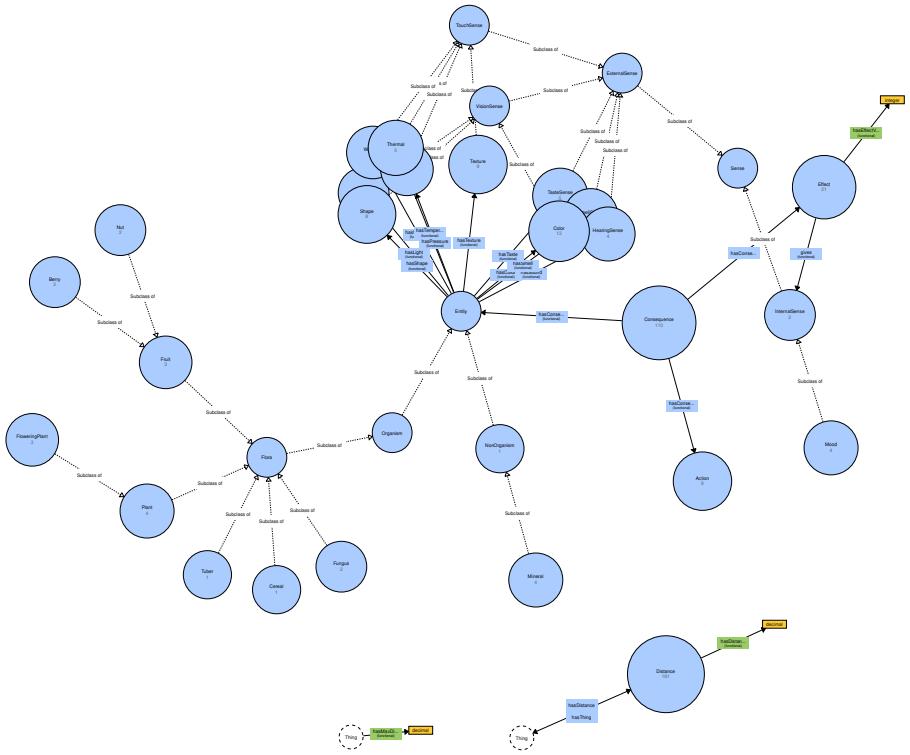

Figure 1: Ontology of the **SymbolicEnv**.

For the reproduction of the results the code can be found on this GitHub repository.

2 The Symbolic Environment: A creature that must survive in the unknown

For the experiments, we decided to create an environment called **SymbolicEnv** which is built on top of an ontology, which allows us to have a standard way of representing our symbolic data (OWL in our case), and to take advantage of the power of symbolic reasoners. Figure 1 graphically illustrates the ontology of the **SymbolicEnv**.

In the **SymbolicEnv**, an agent will have to use its four senses (sight, taste, touch, smell), to choose the most appropriate actions and survive for the longest time, driven by emotions and sensations. To illustrate this, it is possible to identify several parts in the ontology:

Entity The instances that inherit from this class are the entities that the agent may encounter in the environment, e.g. example apple and stone.

ExternalSense The external senses that the agent can perceive, based on these 4 senses, e.g. the texture or colour of an entity.

InternalSense What the agent feels internally, e.g. his mood, his health level or his hunger.

Action Actions that can be taken in the environment, such as eating or attacking.

At each step, the agent receives an observation which is composed of an integer that represents the internal state of the agent, and another integer that represents the entity that the agent encounters.

To create the internal state of an agent, statistics that can evolve naturally over time, or through events, are maintained. Positive stats (energy, health, joy) have a value between 0 and 20 and negative stats (anger, sadness, fear) have a value between 0 and 10. A function allows to discretize the value of a stat into 4 bins and returns an integer between 0 and 3, representing the bin that the value belongs to.

From each positive statistic and the average of the negative ones, it is possible to construct a vector of dimension 4 which represents each discretized value. Since each vector is unique, we can calculate a unique identifier by associating each vector with a value between 0 and $4^4 - 1$.

The **SymbolicEnv** is composed of 6400 observation combinations (256 internal states and 25 entities).

The agent must then choose an action that corresponds to an integer between 0 and the number of action -1, and the agent's statistics are updated accordingly (with an internal logic that is created by reading properties in the ontology).

2.1 Edit distance

The edit distance used in this study quantifies the similarity between two observations, each consisting of an internal state i and an associated entity e. The distance between two observations is defined as the average of the distances between their internal states and their associated entities.

$$d(s_1, s_2) = d((i_1, e_1), (i_2, e_2)) = (d(i_1, i_2) + d(e_1, e_2))/2$$

The distance between two internal states $d(i_1, i_2)$, is calculated by the Euclidean distance normalized distance between 0 and 1 of their two representation vectors.

For the entities, each property p is associated with an *ExternalSense*, and an empirical "distance" is defined between *ExternalSense* of the same type. For example, the distance between yellow and orange might be defined as 1, while the distance between yellow and red might be defined as 2.

To calculate the distance between two entities $d(e_1, e_2)$, the method combines the "tree distance" and "properties distance". The tree distance is defined as the sum of the length of the ancestors of each individual in the ontology. The properties distance is calculated as the sum of the distances between the properties of the two entities, normalized by the maximum distance value of *ExternalSense* of the same type. Finally, the distances are normalized by the maximum distance value.

2.2 Implementation details

The ontology are initially written in the *YAML* format, and translated thanks to Python scripts into the *OWL* format. This allows to automate the creation of some objects in the ontology, because even with very few entities the amount of information to be specified in the ontology becomes very large. We also use *SPARQL* queries to retrieve information efficiently from the database.

The **SymbolicEnv** is built on top of *OpenAI Gym* [1] (a standard API for reinforcement learning) which makes agents testing simple.

3 The Reinforcement Symbolic Agent

The **SymbolicAgent** is built on top of a classical Q-Learning agent, with a modification in the method that updates the Q-table at each step in the environment. In the case of the **SymbolicEnv**, the Q-table is of dimension $256 \times 25 \times 9$ (number of internal states $\times$ number of entities $\times$ number of actions).

The agent takes as argument the distance function to compute the distance between two observations, and for faster computation, distances between each observation are pre-computed and stored in a matrix of size 256^2.

Equation 1 (Symbolic Q-Learning) is implemented in the agent using vectorized operations. Operations likely to be repeated multiple times are stored in the cache.

It will take at least 57600 steps in the **SymbolicEnv** for a classical Q-Learning agent to be updated at least once every value in its Q-Table, assuming that the agent visits a different state and performs differently at each step. The **SymbolicAgent** should have much better generalization capabilities and should need far fewer steps in the environment to perform.

4 Preliminary Results

We trained our agents in the **SymbolicEnv** for 10000 steps, using a linear epsilon greedy policy, and evaluated the agents performances every 10 training steps. The evaluation involved performing 100 rollouts in 100 **SymbolicEnv** with fixed seeds, and the score of an agent was calculated as the sum of the cumulative rewards obtained during each rollout.

All agents have the same hyperparameters, except for the radius ρ which we vary in the case of the **SymbolicAgents**. The experiment consists in observing the differences in performance between a classical Q-Learning agent, and the **SymbolicAgents** with several values of ρ.

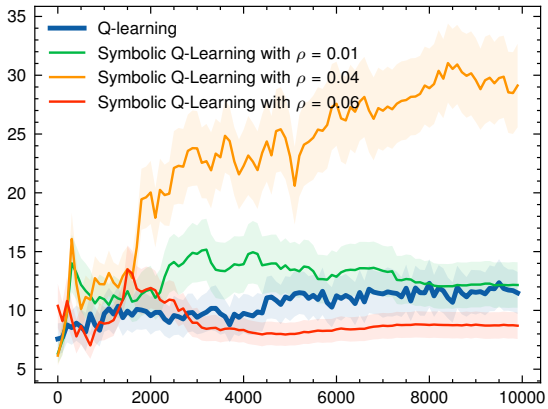

Figure 2: Comparison of Q-Learning and Symbolic Q-Learning Performance with Varying ρ (< 0.1)

Figure 2 shows the evolution of the scores of the different agents throughout the training. Unsurprisingly, the Q-Learning agent struggles to learn, which is simply due to the fact that

the Q-Table is too large, and updating the elements one by one would take much longer to get a good agent. The results of the **SymbolicAgents** are interesting and we can propose several hypotheses:

- With a very small ρ for example (0.01) Symbolic Q-Learning will not update the Q-Table significantly, so the performance will be close to that of classical Q-Learning.
- With a slightly larger radius (0.04) it is possible to obtain performances that are significantly better than the classical Q-Learning, so we see that the ρ hyperarameter is very sensitive, and a small change can drastically change the results.
- When ρ is a bit too large (0.06) the **SymbolicAgent** seems to try to generalise too much, and probably has to update values that it shouldn't, which makes its learning go quite badly, and even performs worse than Q-Learning.

It should be noted here that the results are quite stable, which is not the case with larger radius values as shown in Figure 3.

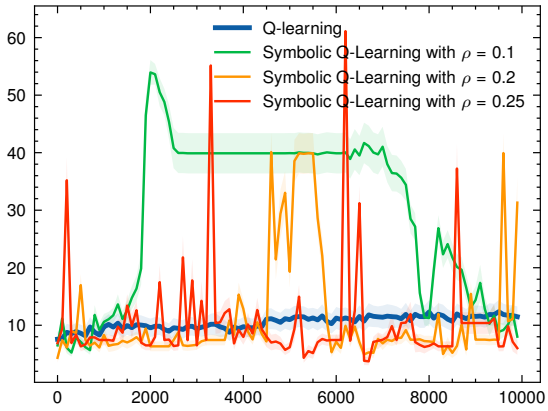

Figure 3: Comparison of Q-Learning and Symbolic Q-Learning Performance with Varying ρ (> 0.1)

Figure 3 shows the same experiment but with agents that have a radius ρ greater than 0.1. The results are very stochastic and difficult to interpret, but it is easy to see that the irregularity is considerable. In these cases many elements of the Q-Table are updated, whether they are relevant or not. This means that the choice of a bad action by the agent can be very penalizing.

These agents are able to reach very high scores early in the learning process, but these are rarely maintained over time.

We can imagine having a variant of **reinforcement symbolic learning** that lowers the radius over time in order to have an agent that will be more and more stable over time

5 Conclusion

In conclusion, we have presented a framework to create a symbolic environment based on an ontology (with OWL), and connect it to an gym environment to make it easier to train agents in it. With this tool, we have created a symbolic environment where an creature needs to adapt to a complex environment in order to survive. We have compared the performance of Q-learning

and symbolic Q-learning with varying radius values in this environment, and the results show that the performance of the symbolic Q-learning approach can be significantly improved with an appropriate choice of the radius parameter.

It is important to note that although the results we obtained with Symbolic Q-Learning are promising, we still do not fully understand how the radius parameter affects the learning process. As we have seen, small changes in the radius value can have a significant impact on the agent's performance and the stability of the learning process, which suggests that the relationship between radius and performance may not be linear. One possible direction for further investigation is to explore ways of dynamically adjusting the radius parameter over time or allowing the agent to learn the optimal radius value. These approaches could potentially improve the adaptability and robustness of the Symbolic Q-Learning algorithm, making it more effective in a wider range of environments.

Despite this, our work has demonstrated the potential of using symbolic environments in reinforcement learning, and we have provided a framework that makes it easy to create symbolic environments using an ontology. We have also shared the Symbolic environment used and the code for our experiments on our open source GitHub repository, which can be used by other researchers and developers to create their own symbolic environments and test new algorithms.

In conclusion, our work highlights the potential benefits of incorporating symbolic reasoning into reinforcement learning and provides a tool for creating symbolic environments that can be used by the research community. The results we obtained with Symbolic Q-Learning suggest that this approach could be a promising direction for future research in the field of RL.

Acknowledgment

We would like to thank the Inria research team Mnemosyne and AEx AIDE for their support during this project. Special thanks go to our supervisors Chloé Mercier, Thierry Viéville, and Axel Palaude for their guidance and expertise, which were instrumental in the development of our work. We also acknowledge their contribution in proposing a version of symbolic Reinforcement Learning that we adapted for our experiments, and without which this project wouldn't have existed.

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[2] Chloé Mercier, Frédéric Alexandre, and Thierry Viéville. Reinforcement Symbolic Learning. In *ICANN 2021 - 30th International Conference on Artificial Neural Networks*, Bratislava / Virtual, Slovakia, September 2021.

[3] Herbert Alexander Simon. The human mind: The symbolic level. In *Proceedings of the American Philosophical Society*, 1993.

[4] Christopher JCH Watkins. Learning from delayed rewards. 1989.

Contents

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399