Towards Automata-Based Abstraction of Goals in Hierarchical Reinforcement Learning

Mehdi Zadem, Sergio Mover, Sao Mai Nguyen, Sylvie Putot

Towards Automata-Based Abstraction of Goals in Hierarchical Reinforcement Learning

Mehdi Zadem*†, Sergio Mover*, Sao Mai Nguyen† and Sylvie Putot*
*LIX, CNRS, École Polytechnique, Institute Polytechnique de Paris, Palaiseau, France
†Flowers Team, U2IS, ENSTA Paris, Institut Polytechnique de Paris & Inria, Palaiseau, France

***Abstract*—Hierarchical Reinforcement Learning (HRL) offers potential benefits for solving long horizon tasks, generally unhandled by standard Reinforcement Learning (RL) techniques, by decomposing the problem and combining simple policies to achieve the goal. They are however still held back by the curse of dimensionality and the ambiguity of selected subtasks. We explore relevant approaches in HRL while highlighting the key challenges of goal representation, high-level planning and propose a research outline tackling them.**

I. Introduction

In the context of learning for robotics, research has achieved remarkable progress in crafting AI systems capable of learning to accomplish tasks in different environments. Standard RL methods are sufficient in simple environments, but issues arise when trying to scale them to high dimensional settings, and environments with sparse rewards. More precisely, some tasks require intermediate sub-tasks to be achieved before the final rewarding result is obtained. They are referred to as hierarchical tasks. A concrete example is Ant Push, where to reach the exit, it is necessary to move the robot so as to go around obstacles at some times and to push obstacles out of the way at other times. Tying the reward only to the distance separating the position of the agent and the exit would not express the underlying sub-tasks necessary for solving the problem. To remedy the sparse reward issues, some pre-engineering is usually integrated to construct a step-by-step reward shaping, and to highlight states presenting interesting properties (these states may correspond to subgoals for example) needed for achieving a final goal. This sacrifices the autonomy of the system and might not always be feasible in case of complex environments. Equipping the agent with a memory structure such as Recurrent Neural Networks (RNNs) to capture long term dependencies across the task was explored but it is often insufficient to represent the underlying structure of the problem and may requires a careful design of the memory structure.

A less hand-crafted solution is therefore required. We focus our attention on Hierarchical Reinforcement Learning which offers an alternative to this tedious approach. In this work, we will examine methods for learning to achieve complex tasks through Hierarchical Reinforcement Learning highlighting key challenges still present in today's state of the art. We also discuss the limitations concerning goal self-generation and the abstraction of the goal space in an effort to optimise learning by composing meaningful and low dimensional policies. We later outline a possible approach that relies on abstracting the goal space and using automata expressed through these subgoals. The automata model a deterministic composition for the high-level behaviour of the agent.

II. Background and related work

To tackle the learning for complex and long tasks, the setting of Hierarchical Reinforcement Learning is a promising approach presenting numerous advantages. The core idea in HRL is to introduce a decomposition mechanism that can solve long horizon tasks by the composition of different simple policies, referred to as primitives. This intuitive process mimics human's cognitive process where we learn simple skills and in combining them we manage to engage in more complex activities. In this section, we explore a set of HRL approaches divided in two categories: autotelic methods that rely on goal self-generation and others guided by instructions.

A. Autonomous Hierarchical Reinforcement Learning

A popular design for HRL algorithms is the Feudal structure first introduced by [1] as a dual structure where two reinforcement learning agents are used: a higher-level agent that selects goals and instructs a lower-level agent to learn for them. [2] is a more advanced instance of this structure relying on advances in Deep Reinforcement Learning and Experience Replay. This algorithm – HIRO – trains two agents to progressively select subgoals leading to the main goal and then learn primitives to accomplish each subgoal. The main appeal of HIRO lies in this decomposition mechanism that leverages several simple sub-problems rather than a long horizon task. This allows the agent to solve the task at hand whereas a standard RL approach cannot converge to an acceptable policy. But despite the impressive results on complex environments – Ant in MuJoCo – the authors still had to resort to manually limiting the observable space to help the agent focus on more interesting regions. Omitting this practice can result in significant performance drop as later examined in [3]. The main criticism here, is that there is no distinct procedure for characterizing optimal subgoals.

Other flavours of this Feudal structure also include FeUdal Networks for Hierarchical Reinforcement Learning (FuN) [4] and Hierarchical DQN (hDQN) [5] that employ a similar Manager-Worker structure with a few key differences among which is the use of a latent state space and then sampling goals from it. The authors make the argument that in addition to the benefit of temporal abstraction, using a latent representation

allows for an appropriate subgoal selection. The conducted experiments highlight the importance of an interpretable and meaningful subgoal space in the global learning. Still, the specific training or design methods for the Manager offer no optimality guarantees with regards to the subgoal representation and selection, and can yield inferior results compared to the performance seen in HIRO. This illustrates the issue where learning state space representation can be inefficient.

B. Hierarchical Reinforcement Learning with instructions

In response to this representation issue, some HRL approaches do indeed sacrifice the full autonomy constraint in favor of more interpretability and efficiency. HAL [6] relies on a set of natural language instructions to guide the hierarchical agents towards accomplishing tasks like object sorting and stacking. The authors in [7] borrow ideas from formal methods to manually model the higher-level agent as an automaton whose transitions correspond to optimal subgoals and accepting path corresponds to the successful behaviour. These elaborate representations highlight the benefits of relying on an abstract goal space to form a set of high-level instructions capable of optimising the decomposition process. It should be pointed out, however, that implementing these approaches requires significant pre-engineering limiting their applicability. DeepSynth [8] partially mitigates this issue by relying only on manually splitting the state space into labeled sets, and learning an automaton for the high-level behaviour through execution traces composed of this high-level abstraction.

In light of these discussions, some key issues are identified; The promise of HRL is still held back by a lack of proper goal space representation. On one hand, using the raw state space as the goal space allows every possible subgoal to be selectable for training primitive policies without loss of information, but it has issues scaling to high dimensions resulting in a sub-optimal high-level policy. On the other hand, using an abstract goal space representation seems to offer potential with regards to tackling complex tasks, and efficiently decomposing the learning problem, but this usually comes at the cost of autonomy. Our aim is to combine the flexibility of HRL with the rigour of symbolic reasoning to construct an algorithm capable of generating an abstract representation for its state space, and to use an interpretable model for instructing the lower-level agent.

III. Solution proposal

Our hypothesis is that integrating an instruction-based component as a higher-level agent in an appropriate Feudal HRL algorithm should allow for a more efficient learning due to better state and subgoal representation. Since DeepSynth[8] offers a way to learn a Deterministic finite automaton (DFA) expressing the desired final behaviour, such a model is a good candidate to act as a higher-level agent. On the other hand, the policy learning in DeepSynth is not Goal-driven but is actually motivated by rewards from the DFA without transmitting an actual goal to the agent. This structure does not address non-stationarity (where the trained primitives evolve and no longer reach the same previous goals creating errors in replay) as discussed by [2] and is less generalisable. Thus the DFA can replace the standard Manager in the HIRO algorithm. HIRO is chosen because it offers flexibility with different state space representations, it mitigates the issue of non-stationarity in HRL and all the primitives are included in a Universal Value Function Approximator (UVFA) policy where the learned goal is a parameter of this function making it easily reusable.

In this setting, let $\mathcal{S}$ be a state space, $\mathcal{A}$ an action space and G a goal such that $G \subseteq \mathcal{S}$. This stipulates that a goal in this instance is a desired observation in the environment. Let us also assume an abstraction function φ that maps states $\mathcal{S}$ to a discrete space $\mathcal{S}^*$. The higher-level agent model is formulated as a DFA $= \{Q, q_0, S^*, F, \delta\}$. with Q a finite set of automaton states that concretely correspond to abstract locations of the program, q_0 an initial state, S^* the alphabet corresponding to the abstract set of goals, F a set of accepting states and $\delta : Q \rightarrow S^*$ a transition function. The desired high-level behaviour of Manager should be in the form of an accepting path, ie a path starting from q_0 and ending in one of the states in F. Launching the episode corresponds to the initial state q_0 of the DFA. From there, the lower-level agent will have to learn to progress from state q_i to state q_{i+1} by following the transition $\delta(q_i)$ which it receives as a goal, and is rewarded depending on how close the reached abstract observation $\varphi(s_{t+1})$ is to $\varphi(q_{i+1})$.

At a preliminary stage, we aim to obtain a reference point as to how efficient this planning system is by using Montezuma's Revenge DFA already developped in [8] and comparing with the original HIRO algorithm. We expect the performance to be on par, if not slightly better, than DeepSynth. In a more advanced stage of the research, we will explore methods for obtaining an abstraction function φ that splits the state space into meaningful subgoals. The challenges involved in this endeavour boil down to providing a clear definition for what a subgoal is with respect to the final goal G, and implementing a method that can identify these splittings. It is expected that the results of this method should fall somewhat shorter than when using a handcrafted abstraction, but superior to the original HIRO algorithm. If we manage to obtain this result, we will have created an effective goal representation learning method.

References

[1] P. Dayan *et al.*, "Feudal reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 5. Morgan-Kaufmann, 1993.

[2] O. Nachum *et al.*, "Data-efficient hierarchical reinforcement learning," in *NeurIPS*, vol. 31. Curran Associates, Inc., 2018.

[3] ——, "Near-optimal representation learning for hierarchical reinforcement learning," in *ICLR*, 2019.

[4] A. S. Vezhnevets *et al.*, "Feudal networks for hierarchical reinforcement learning," *ArXiv*, vol. abs/1703.01161, 2017.

[5] T. D. Kulkarni *et al.*, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Advances in Neural Information Processing Systems*, D. Lee *et al.*, Eds., vol. 29, 2016.

[6] Y. Jiang *et al.*, "Language as an abstraction for hierarchical deep reinforcement learning," *CoRR*, vol. abs/1906.07343, 2019.

[7] X. Li *et al.*, "A formal methods approach to interpretable reinforcement learning for robotic planning," *Science Robotics*, 2019.

[8] M. Hasanbeig *et al.*, "Deepsynth: Automata synthesis for automatic task segmentation in deep reinforcement learning," in *AAAI*, 2021.