Continuously Learning Complex Tasks via Symbolic Analysis (CoLeSlAw)

Abstract

Fully autonomous robots have the potential to impact real-life applications, like assisting elderly people. Autonomous robots must deal with uncertain and continuously changing environments, where it is not possible to program the robot tasks. Instead, the robot must continuously learn new tasks. The robot should further learn how to perform more complex tasks combining simpler ones (i.e., a task hierarchy). This problem is called lifelong learning of hierarchical tasks.

The existing learning algorithm for hierarchical tasks are limited in that: a) they require the robot to execute a large number of real actions to sample the continuous state space of observations, hence requiring a lot of time; b) they cannot deal with subspaces without continuous interpolation, as it is the case for a hierarchy of tasks.

In this Ph.D. project, we will exploit the intuition that set-based reasoning of the continuous space can reduce the number of samples required to learn a hierarchy of tasks and allow for more effective planning of the robot tasks, further handling discontinuities in the task hierarchies. We will design new algorithms to effectively explore the task hierarchies and new reachability algorithms for data-oriented models such as neural networks.

1 Project Description

Context and Problem Definition. In the forthcoming years we expect to adopt robots in several applications like assisting elderly people and maintenance of critical infrastructure (e.g., submarine pipelines). In such contexts, we face a *lifelong learning* [13, 4] problem where we cannot program the tasks a robot can perform in different environment conditions, because those conditions are unknown at design time and/or may change over time. The robot should learn continuously new and complex tasks (i.e., tasks composed of simpler tasks) and adapt them when the environment changes.

Problem Definition. We consider a lifelong learning problem for a single robot interacting in an open environment. The robot knows how to control a set of *primitive actions* $\pi \in \Pi \subset \mathbb{R}^n$ and can obtain *observations* of the environment, $\omega \in \Omega \subset \mathbb{R}^m$, through sensors (e.g., a LIDAR to measure the distance from other obstacles). The mapping between primitive actions and their effects on the environment is unknown a-priori and can change over time. The robot must further learn complex tasks such as *compositional tasks* that are completed by carrying out a sequence of primitive actions (e.g., move an object using a tool). The robot learns such mappings as *forward* or *prediction models* $\mathcal{M}_A : \mathcal{F}_A \rightarrow \Omega_A$ representing the relations between a subset of *features* $\mathcal{F}_A \subset \mathcal{F} = \Pi \cup \Omega$, and subsets of observations $\Omega_A \subset \Omega$. The robot also learns an *inverse* or *control model* $\mathcal{L}_A : \Omega_A \rightarrow \mathcal{F}_A$ to choose which control features can produce a specific observation. During its lifetime the robot maintains a dataset $\mathcal{D}$ of the observations and a hierarchy $\mathcal{H}$ of models, which the robot updates by adding, removing, and changing models.

An example of lifelong learning problem is a 2-wheeled robot moving in a space containing multiple objects of random width and height [9]. The robot continuous action parameter is the electrical current $p \in [-1, 1]$ controlling the wheel's motors, while the observable states are the distances from objects and obstacles, obtained with a LIDAR, and other objects' properties, like color and shape. The robot must learn autonomously different tasks: reach a target position, push and move an object, and push and move an object using a second object as a tool. Each one of these tasks can be achieved only controlling the motors of the robot. Learning the relationship and hierarchy between tasks enables transfer learning and the reuse of knowledge for simple tasks to accomplish the more complex ones. Therefore, curriculum learning (e.g., the robot should learn first how to move and then use this skill to push an object) is fundamental to structure the learning process.

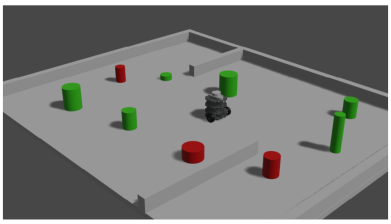

Figure 1: Lifelong learning scenario [9] where a 2-wheeled robot must learn how to move in the space and push objects.

State of the Art. Several *active motor learning* algorithms tackle the continuous learning problem to find a map between the robot motor actions and the observations. In particular, algorithms using intrinsic motivation [10] could successfully guide the autonomous exploration of the environment and build new tasks. The SAGG-RIAC algorithm [1] is an example of such algorithms demonstrating that exploring the feature space, instead of the primitive action space, and guiding the learning process in a goal directed fashion helps to cope with the dimensionality problem. Recent papers [8, 9] introduce CHIME, an algorithm inspired from SAGG-RIAC that learns a task hierarchy. In each iteration, CHIME generates a goal, plans a sequence of primitive or complex tasks from the hierarchy $\mathcal{H}$ that could reach the goal, executes the plan in the physical world, and updates the hierarchy $\mathcal{H}$ according to the observation obtained in the execution step. Despite the benefits derived from being goal oriented and planning a sequence of actions, active learning algorithms such as CHIME still struggle due to the high dimensionality and unboundedness of the problem.

Deep Reinforcement Learning [15, 14] is effective in learning a specific task, also when the set of actions is continuous, but the algorithm requires a fine tuned, task-specific reward function to successfully learn a policy. While this limitation can be relaxed using Universal Value Functions [12] or using intrinsic motivation [10], the experiments in [9] show that hierarchical intrinsic motivation learning algorithms outperform deep reinforcement learning, learning more complex tasks with less iterations. The sub-problem of planning the sequence of tasks tackled in the CHIME algorithm can be solved using *motion planning techniques*, like Rapidly-exploring Random Trees (RRT) [7]. Other works focus on planning in continuous domains [3, 2] and need to explicit model the continuous dynamics withordinary differential equations, while in our settings the models are neural networks constructed from data.

The *symbolic analysis of neural network models* gained interest in the recent years due to their application in autonomous systems (see [11] for a recent survey). These analyses either prove robustness properties (i.e., small perturbations on the inputs of the network only cause small perturbation on the outputs) or functional correctness (i.e., the neural network behaves "as expected"). Symbolic analysis (either using abstract interpretation [5] or constraint solving [6]) can compute sets of forward and/or backward reachable states. One of the main challenge of the analysis problem is in the combinatorial number of states to compute due to the discontinuities of the neural network.

Current Limitations and Challenges. The lifelong learning problem is difficult for several reasons. The problem is *stochastic* (i.e., applying the same action may have different results), *redundant* (i.e., applying different actions may lead to the same result), *high dimensional* (i.e., the state space of the primitive actions and of the observable states is high dimensional), and *unbounded* (i.e., in the time horizon of the mission, the number of actions to execute, and the number of tasks to learn are unbounded). While goal-oriented exploration is an effective strategy to avoid the exploration of the full set of features, only focusing on learning mappings useful to achieve a goal, the intrinsically motivated exploration can handle unbounded and high-dimensional policy spaces, but prove less efficient to explore high-dimensional task spaces, especially to explore all possible dependencies between tasks in hierarchical reinforcement learning: as the number of tasks increases, the space of possible task dependencies increases combinatorially. One of the main limitations of hierarchical reinforcement learning algorithms is that they may still require a large number of iterations to construct the hierarchy $\mathcal{H}$. This curse of dimensionality is all the more costly in robotics that each iteration of the algorithm is expensive and requires the robot to perform a sequence of physical actions (e.g., move in the space or move an object). Each action executed in the "real world" takes time, impacting the algorithm's performance and its adoption to real-case scenarios:

How can we reduce the number of iterations, while still learning the hierarchy of tasks $\mathcal{H}$?

Another limitation is that the existing techniques do not work well on subspaces that do not have continuous interpolation, and in particular the hierarchy $\mathcal{H}$. This happens, for example, when we compare the subtasks needed for a mobile robot to reach a goal position before a wall or behind a wall: in the case the goal position is before the wall, the robot should drive straight to the goal position, whereas in the case the goal position is behind the wall, the robot could need subtasks such as going to the door, opening the door, go past the door then drive straight to the goal position. While these goal positions lie in the same space and are only some centimetres apart, the discontinuity lies not only in the policy parameters, but more fundamentally in the task hierarchy. Unfortunately, hierarchical reinforcement algorithms can not yet handle discontinuities in the task hierarchy, because their approach is based on interpolation of data points and because partitioning task spaces into continuous parts as inputs for the hierarchy $\mathcal{H}$ increases compositionally the number of task dependencies to explore. This is all the more difficult for algorithms such as CHIME where the hierarchy is based on the emergence of control models, which inputs and output spaces are learned through exploration.

How can hierarchies of models take into account discontinuities in the feature space, without combinatorially increasing the number of exploration iterations ?

Scientific Objectives. Our intuition is that reasoning about a single goal and a single sequence of tasks forces the robot to obtain a large amount of data to cover a continuous region of the feature space, and that acquiring such new data in the physical world is expensive. When dealing with a continuous space of features we can instead consider sets of states (e.g., a neighbourhood of the goal) and sets of sequences reaching the goal. That is, reasoning on sets of sequences instead of single sequences will produce higher quality plans, because they will be less affected by redundancy, noise in the measurements and imprecision in the models (i.e., robust) and will consider discontinuities and obstacles. A conjecture in our project is that higher quality plans will reduce the number of iterations (e.g., it will be easier to execute successfully robust plans) to learn a higher number of tasks, and symbolic analysis using sets of spaces can lead to better representations of discontinuities of the environment and tasks, so as to learn discontinuities in the dependencies between tasks.

In this project, we will validate the hypothesis that symbolic analysis reduces the number of iterations required to learn hierarchical tasks, including discontinuous hierarchies.

The first sub-problem we will explore is the *backward reachability analysis in neural networks models*. Given a set of goals $\omega' \subset \Omega_A$ and an inverse model $L_A : \Omega_A \rightarrow \mathcal{F}_A$, this primitive computes the backward reachable set

$\{y \mid \exists x. L_A(x) = y\}$. The main challenge is that L_A is a data-driven model, for example a neural network. In such case, L_A is piece-wise function (e.g., when the network contains ReLU activation functions), representing an exponential number of configurations. For this reason, we will compute *approximations* of such reachable states. While we will evaluate the use of existing reachability analysis techniques for neural networks (see [11]) in our context, we plan to design novel abstract domains to find the right balance between the precision of the approximations and their computational cost. The second sub-problem we solve is to *efficiently search a sequence of tasks* to reach a goal. Here, a challenge is to explore the different orders of compositions of inverse mappings L_A-s. In context like planning, program analysis, and model checking, this challenge is know as exploring different "interleavings". Our problem differs mainly because of the particular structure of L_A and their relationship (i.e., the hierarchy $\mathcal{H}$). We will engineer algorithms to efficiently search this peculiar system and exploit its structure to improve the search (e.g., our continuous reasoning may infer that two inverse models L_{A_1} and L_{A_2} cannot be composed together when starting from a set of features). Finally, we will design a novel active learning framework based on the above primitives. We plan to adopt the architecture of the CHIME algorithm and still rely on intrinsic motivation. However, we expect algorithmic changes after the introduction of symbolic reasoning. For example, we expect changes in the partitioning heuristic of the features and in the selection of target goals.

Methodology, Evaluation Criteria, and Experiments. We will iteratively design different versions of our framework, each one focusing on different problems with increasing challenges and features. This way, we will solve easier problems first, identifying and addressing criticalities in the earlier stages of the project. In each iteration we will design our algorithm, implement it, and then evaluate our research hypothesis on a simulated or real robot (the ASR laboratory already has the robots and facilities to carry out our experiments). We will initially use a simplified version of the case study of shown in Figure 1, which has a simple parameter space (i.e., the motors' urrent) but not considering objects. We will introduce new challenges gradually (e.g., walls, movable objects) and evaluate our approach on simulated robots first and theen real robots. We will then consider a more complex case study involving a robotic arm. We will experimentally compare our algorithm with the existing active learning (e.g., CHIME) and deep reinforcement learning algorithms, comparing the learned hierarchy of tasks and the number of iterations and time required to learn each task.

References

[1] A. Baranes and P. Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics Auton. Syst.*, 61(1), 2013.

[2] S. Bogomolov, D. Magazzeni, S. Minopoli, and M. Wehrle. PDDL+ planning with hybrid automata: Foundations of translating must behavior. In *ICAPS 2015*, 2015.

[3] M. Cashmore, D. Magazzeni, and P. Zehtabi. Planning for hybrid systems via satisfiability modulo theories. *J. Artif. Intell. Res.*, 67, 2020.

[4] Y. Dudai. The Restless Engram: Consolidations Never End. In *Annual Review of Neuroscience*, volume 35. 2012.

[5] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. T. Vechev. AI2: safety and robustness certification of neural networks with abstract interpretation. In *SP*, 2018.

[6] G. Katz, C. W. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *CAV*, 2017.

[7] S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[8] A Manoury, S. M. Nguyen, and C. Buche. CHIME: an adaptive hierarchical representation for continuous intrinsically motivated exploration. In *IRC*, 2019.

[9] A. Manoury, S. M. Nguyen, and C. Buche. Hierarchical affordance discovery using intrinsic motivation. In *HAI*, 2019.

[10] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 2009.

[11] S. Sankaranarayanan, S. Dutta, and S. Mover. Reaching out towards fully verified autonomous systems. In *RP*, 2019.

[12] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *ICML*, 2015.

[13] Sebastian Thrun and Lorien Pratt. *Learning to Learn*. Springer US, Boston, MA, 1998.

[14] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. *CoRR*, 09 2015.

[15] M. Volodymyr, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, Ostrovski G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540), 2015.