Learning Neuro-Symbolic Abstractions for Robot Planning and Learning

Naman Shah

School of Computing and Augmented Intelligence
Arizona State University, Tempe, AZ, USA, 85281
shah.naman@asu.edu

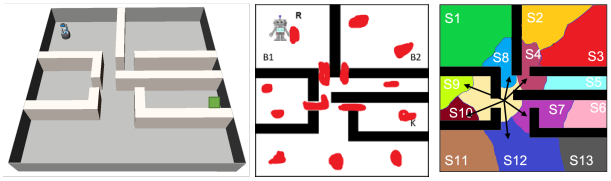
Abstract

Although state-of-the-art hierarchical robot planning algorithms allow robots to efficiently compute long-horizon motion plans for achieving user desired tasks, these methods typically rely upon environment-dependent state and action abstractions that need to be hand-designed by experts. On the other hand, non-hierarchical robot planning approaches fail to compute solutions for complex tasks that require reasoning over a long horizon. My research addresses these problems by proposing an approach for learning abstractions and developing hierarchical planners that efficiently use learned abstractions to boost robot planning performance and provide strong guarantees of reliability.

1 Introduction

Robots needs to plan their actions in order to complete complex tasks in these various areas. E.g., consider the problem shown in Fig. 1(a). However, robot planning over a long horizon is challenging due to the continuous state and action spaces of the robot. Hierarchical approaches (Garrett, Lozano-Pérez, and Kaelbling 2020; Shah et al. 2020) have shown that such abstractions can also be used for efficient robot planning. Unfortunately, these approaches require sound abstractions that are consistent with the motion planning of the robot. However, designing these abstractions is non-intuitive and non-trivial and requires a domain expert. Most related approaches require hand-coded abstractions (Garrett, Lozano-Pérez, and Kaelbling 2020; Shah et al. 2020) or require experience in the test domain (Bagaria and Konidaris 2020) to learn abstractions.

My research aims to answer two crucial research question: (1) Can we automatically learn effective hierarchical state and action abstractions that enable hierarchical planning, and (2) Is it possible to develop efficient approaches that use these automatically generated hierarchical abstractions for robot planning? My research focuses on developing data-driven neuro-symbolic approaches for automatically learning such hierarchical states and action abstractions for complex long-horizon robot planning tasks in unseen environments. I also develop hierarchical planners that use these learned abstractions for efficient robot planning.



(a) (b) (c)

Figure 1: The figure shows the overall approach of my research. (a) shows the ground input motion planning problem. The next step is to identify critical regions as show in (b) and use them synthesize abstract states and actions as shown in (c) using colored cell and arrows respectively.

2 Proposed Approach

My research develops data-driven neuro-symbolic approaches for learning to create hierarchical state and action abstractions for unseen environments. I use the concept of critical regions (Molina, Kumar, and Srivastava 2020) for constructing these hierarchical abstractions. Intuitively, critical regions generalize bottlenecks and hub or access points in an environment in a single concept. I propose to learn a critical region predictor using randomly generated motion plans in a few training environments and use it to automatically identify critical regions in an unseen environment using an occupancy matrix of the environment.

My research uses these automatically identified critical regions to automatically construct a region-based Voronoi diagram (RBVD). A region-based Voronoi diagram partitions the configuration space into different cells. Each cell defines an abstract state inducing an abstraction function. High-level abstract actions are defined as transition between these abstract states induced by Voronoi cells. Fig. 1(c) shows an illustration of a region-based Voronoi diagram. Thus, we obtain a neuro-symbolic atomic abstract representation for an otherwise continuous configuration space of the robot.

Given an abstract representation with a discrete set of abstract states and abstract actions constructed in a bottom-up fashion as outlined above, I focus to develop hierarchical robot planning approaches for efficiently using these abstractions. My research proposes to develop hierarchical probabilistically-complete robot planning algorithms that

interleave high-level symbolic reasoning with continuous low-level motion planning using learned state and action abstractions. Here, an interleaved approach implies that the developed algorithm searches for a high-level abstract plan that has valid low-level refinement for all its symbolic action in iterative setting. This develops a suite of hierarchical algorithms that provide strong theoretical guarantees of probabilistically-completeness and downward refineability.

3 Preliminary Results

This section outlines multiple algorithms for hierarchical planning developed using the above mentioned approach for solving robot planning problem. These approaches include stochastic task and motion planning approach (Sec. 3.1) using hand-coded abstractions and hierarchical planning approaches using learned abstractions (Sec. 3.2 and 3.3).

3.1 Stochastic Task and Motion Planning

Shah et al. (2020) develop an interleaved algorithm for combined task and motion planning. It takes a continuous robot planning problem in the form of a stochastic shortest path (SSP) problem and an entity abstraction as an input and uses it to compute task and motion policy for the input SSP that the robot can execute in the low-level. It iteratively computes a high-level policy and its refinements until it finds a policy that has valid motion planning refinements for all its actions.

The approach is evaluated in multiple settings where combined task and motion planning is necessary to compute feasible solutions. Refining each possible outcome in the policy can take a substantial amount of time. However, ATAM algorithm (Shah et al. 2020) reduces the problem of selecting scenarios for refinement to a knapsack problem and use a greedy approach to prioritize more likely outcomes for refinement. The empirical evaluation shows that this approach allows the robot to start executing action much earlier compared to when actions are selected randomly. Detailed algorithm and experiments are available in the paper.

3.2 Robot Planning Using Learned Abstractions

Shah and Srivastava (2022b) develop a hierarchical planner -- hierarchical abstraction-guided robot planner (HARP) -- that uses automatically synthesized state abstraction in the form of a region-based Voronoi diagram and the action abstractions induced by it. The approach develops a hierarchical planner that uses a multi-source multi-directional variant of the Beam search (Lowerre 1976) for computing a set of high-level plans and use a multi-source multi-directional motion planner LLP (Molina, Kumar, and Srivastava 2020) to simultaneously refine them into a motion plan. Multi-source approaches typically do not work for robot planning. However, critical regions provide crucial information about the states that the robot would potentially visit allowing a multi-source approach to work for robot planning. In summary, Shah and Srivastava (2022b) develop a first approach for learning to create zero-shot state and action abstractions.

The approach is evaluated in multiple settings and compared against state-of-the art sampling-based (Kavraki et al. 1996; LaValle et al. 1998) and learning-based (Molina, Kumar, and Srivastava 2020) motion planners. The results show that using hierarchical planning alongside learning significantly ($\sim 10\times$) improves the efficiency.

3.3 Robot Planning Under Uncertainty

Shah and Srivastava (2022a) develop an approach -- stochastic hierarchical abstraction-guided robot planner (SHARP) -- for computing motion policies for robots in stochastic dynamics. It uses the abstract states defined using an RBVD and defines options that makes transitions between these abstract states. These options are multi-task meaning same set of options can be used for multiple problems in the same environment. SHARP uses A* search to compute a high-level plan by composing options and then uses an off-the-shelf DRL approach to compute policies for these options.

The approach is evaluated in 14 different settings and compared against a re-planning variant of RRT (LaValle et al. 1998), SAC (Haarnoja et al. 2018), and several HRL approaches. While most baselines failed to compute solutions, our approach significantly outperformed all the baselines owing to a dense auto-generated pseudo-reward and an effectively shorter horizon for learning reactive policies.

References

Bagaria, A.; and Konidaris, G. 2020. Option discovery using deep skill chaining. In *ICLR*.

Garrett, C.; Lozano-Pérez, T.; and Kaelbling, L. 2020. PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *ICAPS*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*.

Kavraki, L. E.; Svestka, P.; Latombe, J.-C.; and Overmars, M. 1996. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE TRO*, 12(4).

LaValle, S. M.; et al. 1998. *Rapidly-exploring random trees: A new tool for path planning*. Iowa State University.

Lowerre, B. T. 1976. *The Harpy Speech Recognition System.* Carnegie Mellon University.

Molina, D.; Kumar, K.; and Srivastava, S. 2020. Identifying Critical Regions for Motion Planning using Auto-Generated Saliency Labels with Convolutional Neural Networks. In *ICRA*.

Shah, N.; Kala Vasudevan, D.; Kumar, K.; Kamojjhala, P.; and Srivastava, S. 2020. Anytime Integrated Task and Motion Policies for Stochastic Environments. In *ICRA*.

Shah, N.; and Srivastava, S. 2022a. Multi-Task Option Learning and Discovery for Stochastic Path Planning. *arXiv preprint arXiv:2210.00068*.

Shah, N.; and Srivastava, S. 2022b. Using Deep Learning to Bootstrap Abstractions for Hierarchical Robot Planning. In *AAMAS*.