Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Doctoral Consortium Track

Reliable Neuro-Symbolic Abstractions for Planning and Learning

Naman Shah
Arizona State University,
Tempe, AZ, USA
namanshah@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 [Dantam *et al.*, 2018; Garrett *et al.*, 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 [Dantam *et al.*, 2018; Garrett *et al.*, 2020; Shah *et al.*, 2020] or require experience in the test domain [Kurutach *et al.*, 2018; Bagaria and Konidaris, 2020] to learn abstractions.

Through my work, I aim 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.

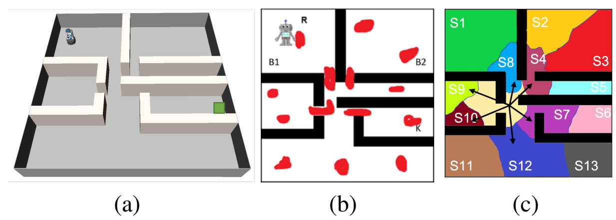

Figure 1: The figure shows the overall approach of my research. (a) shows the ground input motion planning problem. The next step is to identify critical regions as show in (b) and use them synthesize abstract states and actions as shown in (c) using colored cells and blacked arrows respectively. These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

2 Proposed Approach

My research develops data-driven neuro-symbolic approaches for learning hierarchical state and action abstractions. It uses a set of training environments and a collection of motion plans in these training environments for learning a predictor that identifies *critical regions* [Molina *et al.*, 2020] in a new environment. Formally, critical regions are defined as regions that has high density of solutions for a given class of problems but difficult to sample under a uniform probability distribution. Intuitively, critical regions generalize the concept of pinch points and hub points in the environments. This research aims to learn a predictor that learn to automatically identify critical regions in unseen test environments.

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 in different cells. Each cell defines an abstract state inducing an abstraction function. High-level abstract actions can be defined as transition between these Voronoi cells. Fig. 1(b) shows an illustration of a region-based Voronoi diagram. States in the input problem are defined as configurations of the robot and actions are defined as changes in the configuration of the robot. However, the abstract states at the high-level are atomic and actions are defined as transitions between these cells.

Once these abstract states and actions are identified, they can be used with hierarchical algorithms to compute solution for the input robot planning problem. My research proposes to develop an hierarchical interleaved probabilistically-complete algorithm that uses learned state and action abstractions. It should use abstract states and actions for computing high-level plans and then use the low-level descriptions to refine these high-level actions into motion plans that the robot can execute in the real world.

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 *et al.*, 2020] to simultaneously refine them into a motion plan. While multi-source approaches typically do not work for robot planning. However, critical regions allow a multi-source approach to work as it identifies states the robot would potentially visit.

The approach is evaluated in multiple settings and compared against state-of-the art sampling-based [Kavraki *et al.*, 1996; LaValle and others, 1998; Kuffner and LaValle, 2000] and learning-based [Molina *et al.*, 2020] motion planners. The results show that using hierarchical planning alongside learning significantly (about $10\times$) 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 and others, 1998], SAC [Haarnoja *et al.*, 2018], and several HRL approaches. While most HRL methods failed to compute solutions, our approach significantly outperformed all the baselines.

References

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

[Dantam *et al.*, 2018] N. Dantam, Z. Kingston, and S. Chaudhuriand L. Kavraki. An incremental constraint-based framework for task and motion planning. *IJRR*, 37(10):1134–1151, 2018.

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

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

[Kavraki *et al.*, 1996] LE. Kavraki, P. Svestka, J-C Latombe, and MH. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE TRO*, 12(4), 1996.

[Kuffner and LaValle, 2000] JJ. Kuffner and SM. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *ICRA*, 2000.

[Kurutach *et al.*, 2018] T. Kurutach, A. Tamar, G. Yang SJ. Russell, and P. Abbeel. Learning plannable representations with causal InfoGANs. In *NeruIPS*, 2018.

[LaValle and others, 1998] Steven M LaValle et al. *Rapidly-exploring random trees: A new tool for path planning*. Iowa State University, 1998.

[Lowerre, 1976] BT. Lowerre. *The Harpy Speech Recognition System.* Carnegie Mellon University, 1976.

[Molina *et al.*, 2020] D. Molina, K. Kumar, and S. Srivastava. Identifying critical regions for motion planning using auto-generated saliency labels with convolutional neural networks. In *ICRA*, 2020.

[Shah and Srivastava, 2022a] N. Shah and S. Srivastava. Multi-task option learning and discovery for stochastic path planning. *arXiv preprint arXiv:2210.00068*, 2022.

[Shah and Srivastava, 2022b] N. Shah and S. Srivastava. Using deep learning to bootstrap abstractions for hierarchical robot planning. In *AAMAS*, 2022.

[Shah *et al.*, 2020] N. Shah, D. Kala Vasudevan, K. Kumar, P. Kamojjhala, and S. Srivastava. Anytime integrated task and motion policies for stochastic environments. In *ICRA*, 2020.