A Novel Abstraction Framework for Online Planning (Extended Abstract)

Ankit Anand, Aditya Grover, Mausam and Parag Singla
Indian Institute of Technology, Delhi
New Delhi, India
{ankit.anand, adityag.cs111, mausam, parags}@cse.iitd.ac.in

ABSTRACT

Abstractions are a useful tool for computing policies in large domains modeled as a Markov Decision Process. Prior work in this field is mostly focused on developing different notions for state abstractions. In this paper, we develop a novel framework for abstractions, which unifies prior work and directly exploits symmetry at the state-action pair level, thereby uncovering a much larger number of symmetries in a given domain. We describe the application of abstractions computed through this framework in UCT, a popular MCTS technique for online planning.

Categories and Subject Descriptors

K.3.3 [**Computing methodologies**]: Artificial intelligence Planning and scheduling

General Terms

Algorithms

Keywords

Abstraction, MDPs, UCT

1. INTRODUCTION

Markov Decision Processes are an attractive framework for sequential decision making under uncertainty. However, classical MDP algorithms operate in flat state and action spaces and become intractable when the state space becomes too large. In many real world scenarios, a state is composed of a number of features and hence, state space grows exponentially with number of features leading to inapplicability of traditional planning algorithms.

One approach to address this problem is to exploit symmetries in a given domain. In many scenarios, states are not completely independent of each other – several states may be symmetrical for the purpose of planning and can be abstracted out. For example, a robot's available hard-drive may be irrelevant for the purpose of navigation leading to several symmetric states. Similarly, various actions may be symmetric too – if the goal is to reach the center of the hall, the right action taken in the left corner will be symmetric to left action taken in the right corner. Prior work in state abstractions for offline MDP algorithms [2, 6] aggregates similar states using local transition probabilities and associated costs.

In recent times, sampling-based techniques such as Monte Carlo Tree Search (MCTS) algorithms have gained popularity in the context of online planning, especially for large domains. These algorithms are independent of the size of the state-space and have anytime property. A popular MCTS variant - UCT [5] has enjoyed tremendous success in recent times in various areas of decision making such as planning under uncertainty , reinforcement learning and game playing. While UCT has become an algorithm of choice, it operates in flat space and the policies obtained are approximate; their quality depends on the extent to which UCT exploits the explored state space in the given time frame.

While both these threads of research (MCTS and abstractions) are well developed, there is only very recent work that begins to combine the two. Hostetler et.al. [3] defines a series of theoretical principles for state abstractions in a sampling-based algorithm. They do not provide any automated algorithm to compute abstractions. Probably the closest to our work is Jiang et.al. [4]. They provide an algorithm that operates within the UCT framework and computes state abstractions based on the definitions proposed by Givan et. al. [2]. Our experiments show that these state abstractions are not much effective in practice.

In this context, we propose a novel abstraction framework that exploits symmetries at the *state-action pair* level in addition to state abstractions. State-action pairs are equivalent if they transition to equivalent states with same probabilities and immediate costs. Our experiments reveal that we can utilize many more state-action pair abstractions even where there are no state abstractions. In addition to directly benefiting from state-action pair symmetries, our framework unifies the previously proposed notions for state abstractions. We develop an MCTS-based algorithm that makes use of this framework. Experiments on a wide variety of benchmark domains reveal significant gains over previous approaches.

Overall this paper makes two principle contributions:

1. We propose a novel framework for state and state-action pair abstractions. This framework subsumes all previously defined notions of state abstractions.
2. We develop an algorithm to use this framework within UCT. Empirical evaluations on benchmark domains highlight the promise of our framework.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

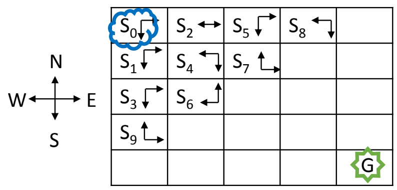

Figure 1: Modified Sailing Wind. The arrows in each grid cell point to the permitted action directions.

2. ABSTRACTION FRAMEWORK

To understand our abstraction framework, it is helpful to visualize an MDP as a bipartite graph of state and action nodes. While the graph contains only one copy for every state node, same actions might apply for different states and hence, to uniquely identify action nodes, we consider them as a pair of state and actions. With this representation, we now give a mutually recursive formulation for our abstraction framework. We say that two state-action nodes are equivalent if and only if they transition to equivalent states with same probabilities and incur equal costs. Similarly, two states are equivalent if and only if we can define a mapping of their applicable actions leading to equivalent state-action pair nodes. We illustrate the application of our abstraction framework in UCT with a simple example.

Example: In a deterministic version of the sailing wind domain [5] depicted in Figure 1, the agent starts from S_0 and its goal is to navigate to the bottom right corner of the grid by selecting either of the actions-N, S, W, E directions. The permitted actions owing to wind direction are indicated by the arrows in Figure 1. The state of the agent is modeled by the current grid cell coordinates of the agent and the corresponding wind direction.

Figure 2 shows the UCT tree for the domain instance under consideration. Since UCT is applied in finite-horizon settings, we consider state (and state-action pair) nodes separately at different depths. We first describe how Jiang et al.'s [4] algorithm would work on the above sampled tree. The algorithm computes abstractions explicitly only in the state layers. At the lowest level ($d = 3$), all the nodes would be combined together into one abstract state since there are no further transitions from them. At one level above ($d = 2$), S_3 and S_5 will be combined together since both of them have the same set of actions (E and S) and these actions transition into same equivalence class states at the lowest level. At each level, the equivalent state partitions produced by this method are shown using the dotted lines. Note that if we apply Ravindran et al's [6] notion of state equivalence, we obtain additional state symmetries (indicated in red) such as S_4 belonging to the same equivalence class as S_3 and S_5.

In our approach, in addition to finding abstractions at the state layers, we also explicitly work at the state-action layers. Intuitively, in our algorithm the grouping of state layer nodes depends on the transitions into the state-action layer nodes below them. Similarly, the grouping of state-action layer nodes depends on the transitions into the state layer nodes following them. Equivalent state-action pair nodes at $d = 2$ and $d = 3$ are indicated in green in Figure 2. The reader can refer to [1] for details of the proposed framework.

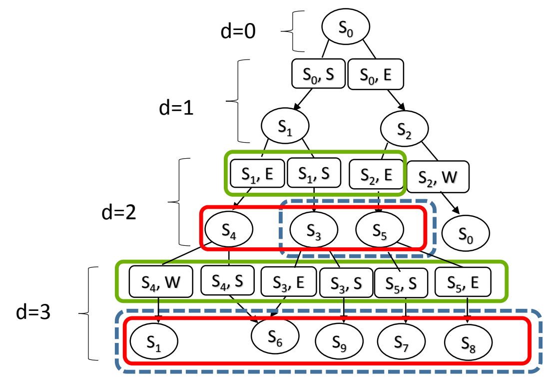

Figure 2: Monte-Carlo Search Tree for Sailing Wind. Previous notions of abstractions are only able to capture symmetries at the state level (blue and red), whereas our framework discovers additional symmetries at the state-action pair level (green).

3. EXPERIMENTAL EVALUATION

We evaluated our framework of state and state-action pair abstractions on a variety of benchmark domains viz. Sailing Wind, Navigation etc.. The experiments revealed that our abstraction framework shows upto 26% improvements in policies over existing notions of abstraction viz. Givan et. al. [2] and Ravidran et al. [6] based on Jiang et al.'s [4] implementation in UCT framework.

4. CONCLUSION AND FUTURE WORK

This paper defines a new class of state-action pair abstractions and incorporates these within a new framework which subsumes past work on state and action abstractions in MDPs [2, 4, 6]. In future, we would like to explore the theoretical and scalability aspects of our proposed framework in greater detail and extend the applicability of our work to factored representations. Extending our ideas to other decision-theoretic frameworks such as reinforcement learning is also a subject of future work.

REFERENCES

[1] A. Anand, A. Grover, Mausam, and P. Singla. A Novel Abstraction Framework for Online Planning. Technical report, Indian Institute of Technology, Delhi, 2015.

[2] R. Givan, T. Dean, and M. Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1-2):163 – 223, 2003.

[3] J. Hostetler, A. Fern, and T. Dietterich. State Aggregation in Monte Carlo Tree Search. In *AAAI*, 2014.

[4] N. Jiang, S. Singh, and R. Lewis. Improving UCT Planning via Approximate Homomorphisms. In *AAMAS*, pages 1289–1296, 2014.

[5] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, pages 282–293. Springer, 2006.

[6] B. Ravindran and A. Barto. Approximate homomorphisms: A framework for nonexact minimization in Markov decision processes. In *Proc. 5th Int. Conf. Knowledge-Based Computer Systems*, 2004.