Monte Carlo Hierarchical Model Learning

(Doctoral Consortium)

Jacob Menashe and Peter Stone
The University of Texas at Austin
Austin, Texas
{jmenashe,pstone}@cs.utexas.edu

ABSTRACT

Reinforcement learning (RL) is a well-established paradigm for enabling autonomous agents to learn from experience. To enable RL to scale to any but the smallest domains, it is necessary to make use of abstraction and generalization of the state-action space, for example with a factored representation. However, to make effective use of such a representation, it is necessary to determine which state variables are relevant in which situations. In this work, we introduce T-UCT, a novel model-based RL approach for learning and exploiting the dynamics of structured hierarchical environments. When learning the dynamics while acting, a partial or inaccurate model may do more harm than good. T-UCT uses graph-based planning and Monte Carlo simulations to exploit models that may be incomplete or inaccurate, allowing it to both maximize cumulative rewards and ignore trajectories that are unlikely to succeed. T-UCT incorporates new experiences in the form of more accurate plans that span a greater area of the state space. T-UCT is fully implemented and compared empirically against B-VISA, the best known prior approach to the same problem. We show that T-UCT learns hierarchical models with fewer samples than B-VISA and that this effect is magnified at deeper levels of hierarchical complexity.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents

General Terms

Algorithms

Keywords

Single and multi-agent learning techniques; Reinforcement Learning; Factored Domains; Model Learning; Hierarchical Skill Learning; Monte Carlo Methods

1. INTRODUCTION

Suppose you are tasked with driving to a new supermarket downtown. At short notice you might be able to come up with some simple instructions, such as "head south for approximately 3 miles." Before actually making the trip you could consult a map and look at a couple of possible routes, then settle on the route that seems the best given the distance, the time of day, etc. Finally you try out your selected route and use this new experience to help you plan better in the future.

In further detail, the process of planning your actions is divided into distinct phases. The first phase is target selection, in which you decide on the supermarket as your destination. The next is a rough planning phase, in which you select a high-level action sequence to consider: "head south for 3 miles." For the third phase you then simulate the experience of navigating to your target by looking at a map and planning out the specific roads you'll be taking. Finally you execute an action sequence by following your planned route to the new supermarket.

In this work we introduce an implementation of this approach to model-based planning, namely *Transition-based Upper Confidence Bounds for Trees*, or T-UCT. We draw from the widely successful UCT algorithm [5] by extending it for use with action sequences rather than primitive actions. This extension allows us to make long-term, compound planning decisions that respect both the intermediate reward and transition dynamics of a given environment. T-UCT selects targets to explore novel areas of the state space, performs randomized depth-first graph search for rough planning, and then uses UCT to carry out Monte Carlo simulations. Finally, T-UCT executes the best plan derived from this process to explore the environment.

2. PROBLEM DEFINITION

In this work we tackle the problem of learning a conditional environment model from scratch in an intractably complex environment. Moreover, we iteratively leverage learned environment dynamics to guide exploration and learn more efficiently. We use a flexible combination of random target selection, discrete planning, and traditional Reinforcement Learning (RL) methods to allow an agent to both learn from experience and navigate through a complex environment.

We model the environment as a Factored Markov Decision Process (FMDP). An FMDP is defined by a tuple $M = \langle \mathcal{S}, A, P, R \rangle$ where $\mathcal{S} = S_1 \times S_2 \times \cdots S_n$ for some n. Here $\mathcal{S}$ represents our *factored* state space, with each S_i representing a state variable of $\mathcal{S}$. A is the action space, representing the set of all actions available in M. P and R are the transition and reward functions, respectively. Here

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.*

$P(s, a, s') = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$ for $s, s' \in \mathcal{S}$ and $a \in A$, and $R(s, a, s') \in \mathbb{R}$.

Our problem thus becomes one of modeling P. That is, we wish to find a compact representation of P in order to predict the value $P(s, a, s')$, for all possible s', given $s, a \in \mathcal{S} \times A$.

3. RELATED WORK

There has been much focus on hierarchical reinforcement learning [1, 8, 7], as well as using factored domains to simplify state representations [4, 3]. Our algorithm has been influenced by recent work that seeks to leverage factored representations to create hierarchical options [2, 8]. Parallel research has focused on minimizing extraneous exploration through the use of Monte Carlo Tree Search, most notably with the UCT algorithm [5].

Our work combines these approaches and applies them to hierarchical model learning. We use the compact factored model learning of Jonsson and Barto [3] along with the hierarchical bootstrapping techniques applied by Vigorito and Barto [8]. We use a hierarchically enhanced version of UCT similar to Vien and Toussaint [7]. The result is an algorithm that produces compact, factored models from scratch and uses them to efficiently explore otherwise intractably large state spaces with complex environment dynamics.

4. PUBLISHED WORK

We have one published paper on this topic [6] to be presented at AAMAS'15. In this work we present an approach to the stated problem using UCT as the basis for action evaluation and selection. We iteratively build and query an environment model consisting of a collection of Dynamic Bayesian Networks, as is done in previous work [2, 8]. We use these models to generate *Transitions*: tuples each consisting of a single primitive action as well as predictive information derived from our learned model. We use transitions to inform our UCT-based action selection process with model-specific information. We call this enhanced UCT-based method *Transition UCT* (T-UCT).

We find that our algorithm outperforms the best known solution to our evaluated domains, which we refer to as B-VISA. We evaluate our findings on both the original domain provided to illustrate B-VISA's efficiency [8], as well as our own complexified version of this domain. In both cases we show significant improvements in sample efficiency, finding that our method fully and accurately learns environment dynamics more quickly due to its ability to simulate experiences and integrate flexible planning.

5. FUTURE WORK

An obvious limitation of our algorithm is the reliance on discrete state and action representations, necessitating limited or discretized environments. Our immediate future work will focus on integrating the discrete planning enabled by T-UCT with low-level controllers in continuous domains.

Additionally, both B-VISA and T-UCT rely on iterating all state variables and actions in the domain and modeling conditional dependencies between variables that may be completely unrelated. As a simple example, consider the problem of predicting the color of a traffic light in Texas based on the weather in London. Clearly these two variables can be ignored with respect to one another, however we must find a way to sanely ignore these potential associations in order to scale to large or infinite state-action spaces.

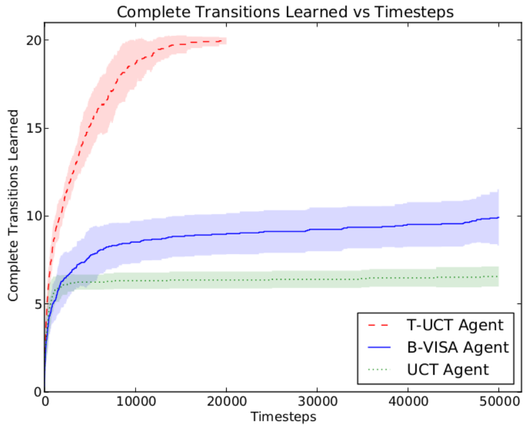

Figure 1: A comparison of T-UCT, B-VISA, and UCT on the random lights domain [6]. The data show the number of timesteps required for the agents to learn correct transitions for all of the 20 lights in the domain, averaged over 25 trials. Shaded regions represent standard error. Higher values at each timestep are better. The results show that T-UCT consistently outperforms both B-VISA and UCT. These results are significant with $p < .001$.

After making these improvements to T-UCT we have a variety of possible directions. The ultimate goal for this work is to enable a robotic agent to learn hierarchical tasks in a real-world environment, such as driving a car or playing a sport. We plan to coordinate our learning mechanisms with other work in the areas of computer vision and perception in order to realize this work in a physical domain.

REFERENCES

[1] T. G. Dietterich. The maxq method for hierarchical reinforcement learning. In *ICML*, pages 118–126. Citeseer, 1998.

[2] A. Jonsson and A. Barto. Causal graph based decomposition of factored mdps. *The Journal of Machine Learning Research*, 7:2259–2301, 2006.

[3] A. Jonsson and A. Barto. Active learning of dynamic bayesian networks in markov decision processes. In *Abstraction, Reformulation, and Approximation*, pages 273–284. Springer, 2007.

[4] M. Kearns and D. Koller. Efficient reinforcement learning in factored mdps. In *IJCAI*, volume 16, pages 740–747, 1999.

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

[6] J. Menashe and P. Stone. Monte carlo hierarchical model learning. In *To appear in AAMAS*, 2015.

[7] N. A. Vien and M. Toussaint. Hierarchical monte-carlo planning. 2015.

[8] C. M. Vigorito and A. G. Barto. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development*, 2(2):132–143, June 2010.