

Task Scoping for Efficient Planning in Open Worlds (Student Abstract)

Nishanth Kumar,* Michael Fishman,* Natasha Danas,
Stefanie Tellex, Michael Littman, George Konidaris

Brown University Department of Computer Science
115 Waterman Street, Providence, RI 02912
nishanth_kumar@brown.edu, michaeljfishman@gmail.com

*

Abstract

We propose an abstraction method for open-world environments expressed as Factored Markov Decision Processes (FMDPs) with very large state and action spaces. Our method prunes state and action variables that are irrelevant to the optimal value function on the state subspace the agent would visit when following any optimal policy from the initial state. This method thus enables tractable fast planning within large open-world FMDPs.

Introduction

In large, open-world scenarios, intelligent agents will be faced with environments described by a prohibitively large number of state and action variables. However, most of these variables will be irrelevant to any specific goal. A general-purpose agent that must solve many different tasks in such open worlds cannot rely on hand-curated compact models, but is also unlikely to be able to plan in the presence of large numbers of irrelevant variables. It must therefore discern the relevant aspects of the environment on its own.

Factored Markov Decision Processes (FMDPs) offer a structured formalism to model such large sequential decision-making problems. Existing algorithms have attempted to exploit this structure in the transition and reward functions of FMDPs to perform abstraction (Boutillier, Dearden, and Goldszmidt 1995) and thus avoid the combinatorial explosion from planning in large FMDPs. However, even these methods often fail at open-world scale.

We propose a novel abstraction algorithm called *task scoping* for FMDPs augmented with a start condition. Task scoping abstracts away any state variables that either 1) cannot affect any state variables mentioned in the reward function or 2) guaranteed to have the “correct” value throughout the agent’s trajectory, and so can be ignored.

Our algorithm then returns a ‘scoped’ FMDP with state and action spaces that only contain variables and actions relevant to the current task. The state space size is at most the same as the original, but in most cases is much smaller. Any

*Corresponding authors equally contributed; others co-advised. Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

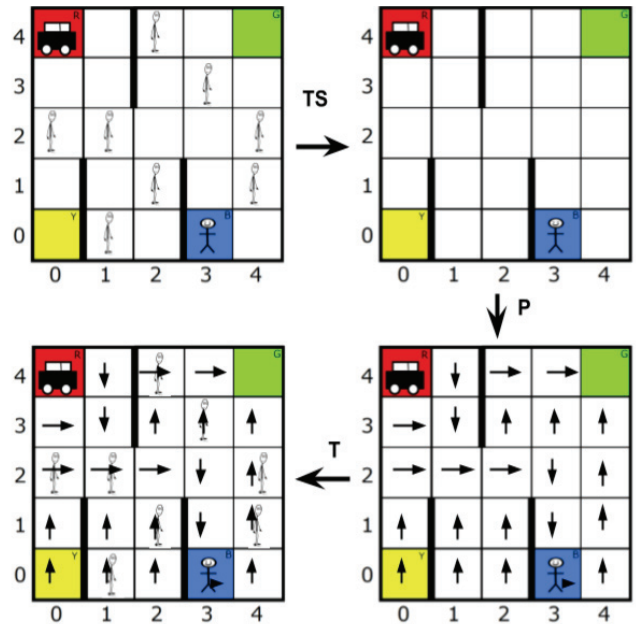


Figure 1: Our pipeline for a variant of Dietterich (2000)’s Taxi Domain augmented with irrelevant passengers. The original FMDP undergoes Task Scoping (TS) to remove all irrelevant passengers (not in the blue square). A planner performs Planning (P) to compute a policy in the scoped FMDP. Finally the policy is Transferred (T) to the original FMDP.

FMDP planner can be used to compute a policy in this abstract FMDP, and this policy can then be directly transferred to the original FMDP.

Methodology

Task scoping (TS in Figure 1, pseudo-code in Algorithm 1) performs a form of temporally-abstracted least commitment planning (Weld 1994), via goal-regression: monitoring whether each new sub-goal has a causal-link from the start condition (PROCESS_CONDITIONS), and whether each new action threatens existing causal-links (FIND_THREATS). The intuition being— if a goal is

attainable— there exists some sequence of actions, that either affects the goal condition directly, or affects preconditions of actions that affect the goal condition, and so on. Our algorithm alternates between PROCESS_CONDITIONS and FIND_THREATS, reducing the FMDP state and action space, without actually planning or rendering the optimal policy infeasible. The current implementation uses the theorem prover z3 to check whether the start condition implies each condition (De Moura and Bjørner 2008).

Algorithm 1 All defined variables are taken in as arguments.

```

1: procedure SCOPE_TASK( $s_0, g, actions$ )
2:    $discovered\_conds = g.split\_conjuncts()$ 
3:    $agenda = g.split\_conjuncts()$ 
4:    $used\_actions = []$ 
5:   while not  $agenda.empty()$  do
6:      $PROCESS\_CONDITIONS()$ 
7:      $FIND\_THREATS()$ 
8:      $relevant\_conds = discovered\_conds - causal\_links$ 
9:      $relevant\_state\_vars = \bigcup_{c \in relevant\_conds} c.vars$ 
10:    return ( $relevant\_state\_vars, used\_skills$ )
11: procedure PROCESS_CONDITIONS
12:    $cond = agenda.pop()$ 
13:   if  $s_0 \implies cond$  then
14:      $causal\_links.push(cond)$ 
15:   else
16:     for  $a \in actions$  do
17:       if  $a.target\_vars \cap cond.vars \neq \emptyset$  then
18:         for  $c \in a.prec.split\_disj()$  do
19:           if  $c \notin discovered\_conds$  then
20:              $discovered\_conds.push(c)$ 
21:              $agenda.push(c)$ 
22: procedure FIND_THREATS
23:    $broken\_links = []$ 
24:   for  $causal\_link \in causal\_links$  do
25:     for  $action \in used\_actions$  do
26:       if  $action.possibly\_affected\_vars \cap$ 
27:          $causal\_link.vars \neq \emptyset$  then
28:          $broken\_links.push(causal\_link)$ 
29:   for  $causal\_link \in broken\_links$  do
30:      $causal\_links.remove(causal\_link)$ 

```

Related Work

Dietterich (2000) and others have researched learning a state abstraction in a reinforcement learning context (Andre and Russell 2002). We focus on state abstraction in the FMDP planning context, which enables more effective abstraction.

Boutillier, Dearden, and Goldszmidt (1995) and others researched using the Dynamic Bayes Net (DBN) of an FMDP to structure the value and policy functions as algebraic decision diagrams (Hoey et al. 2013). The abstraction in these works cannot be separated from the planning: they must check whether each variable is relevant at each iteration.

Helmert (2006) introduced Fast-Downward, a classical planning algorithm, that ignores trivially irrelevant state variables, but does not exploit the start condition to prune non-trivially irrelevant state variables. Additionally, it uses

a heuristic to break cycles in the causal-graph, while our approach has no heuristics. Muise, Mcilraith, and Christopher Beck (2012) use Fast-Downward on a determined version domain to find a compact state representation.

Squire and desJardins (2016) propose a method to create an abstraction of a partially-observable OO-MDP that ignores irrelevant state variables; however, this performs an exhaustive enumeration of all possible abstractions, which would be computationally intractable for large state-spaces.

Conclusion and Future Work

We propose task scoping, which leverages knowledge of an agent’s initial state, to prune states irrelevant to the optimal value function from the FMDP. Preliminary experiments on large multi-passenger taxi-domains indicate that our method enables state-of-the-art FMDP planners to solve otherwise-intractable problems. In the future, we hope to test our algorithm on all International Planning Competition domains and truly ‘open-world’ video game or robotics domains expressed as massive partially-observable FMDPs.

References

- Andre, D., and Russell, S. J. 2002. State abstraction for programmable reinforcement learning agents. In *Eighteenth National Conference on Artificial Intelligence*, 119–125. Menlo Park, CA, USA: American Association for Artificial Intelligence.
- Boutillier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting structure in policy construction. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI’95*, 1104–1111. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- De Moura, L., and Bjørner, N. 2008. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS’08/ETAPS’08*, 337–340. Berlin, Heidelberg: Springer-Verlag.
- Dietterich, T. G. 2000. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Int. Res.* 13(1):227–303.
- Helmert, M. 2006. The fast downward planning system. *J. Artif. Int. Res.* 26(1):191–246.
- Hoey, J.; St-Aubin, R.; Hu, A. J.; and Boutillier, C. 2013. SPUDD: stochastic planning using decision diagrams. *CoRR* abs/1301.6704.
- Muise, C.; Mcilraith, S.; and Christopher Beck, J. 2012. Improved non-deterministic planning by exploiting state relevance. *ICAPS 2012 - Proceedings of the 22nd International Conference on Automated Planning and Scheduling*.
- Squire, S., and desJardins, M. 2016. Abstracting complex domains using modular object-oriented markov decision processes. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16*, 4264–4265. AAAI Press.
- Weld, D. S. 1994. An introduction to least commitment planning. *AI Magazine* 15(4):27–61.