Task Scoping for Efficient Planning in Open Worlds

Nishanth Kumar*, Michael Fishman*, Natasha Danas,
Stefanie Tellex, Michael Littman and 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 leverages knowledge of an agent's initial state to discover and prune states and actions that are irrelevant to the optimal value function given 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, almost all of these variables will be irrelevant to any specific objective. A general-purpose agent that may be asked to solve many different tasks in such open worlds cannot rely on hand-curated compact models, but is also unlikely to be able to successfully plan in the presence of large numbers of irrelevant state 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 (Boutilier, 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) are 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 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.

*Corresponding authors equally contributed; others co-advised.

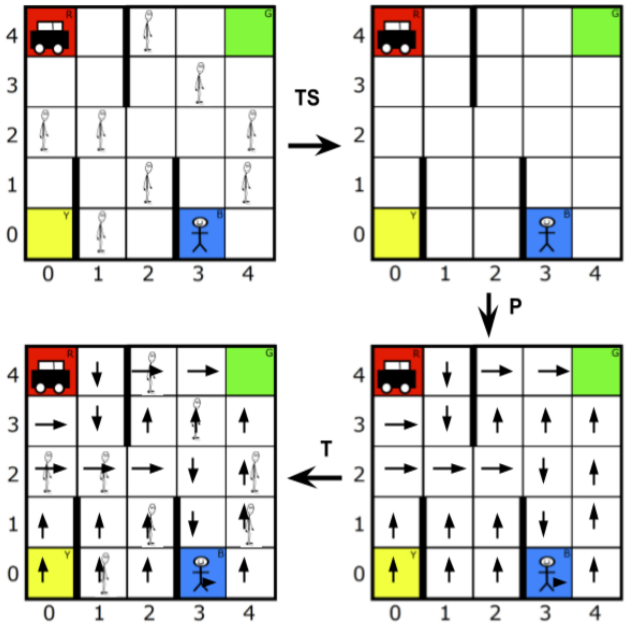

Figure 1: An overview of our pipeline for a variant of Dietterich (2000)'s popular Taxi Domain augmented with irrelevant passengers. The original multi-passenger taxi FMDP undergoes Task Scoping (**TS**) to remove all irrelevant passengers (who are not in the blue square), a planner performs Planning (**P**) to compute a policy in the smaller, scoped FMDP, and finally the policy is Transferred (**T**) to the original FMDP.

Methodology

Our task scoping algorithm is a kind of temporally-abstracted least commitment planning (Weld 1994). It performs goal-regression while monitoring whether each new sub-goal has a causal-link from the start condition, and whether each new action threatens existing causal-links. The temporal-scoping comes from considering *all* potentially-useful actions when backchaining, rather than a single action. The intuition is that, if a goal is attainable, it must be attainable by taking some sequence of actions that affect the goal condition directly, or that affect preconditions of actions that affect the goal condition, and so on. Thus, our Task

Scoping algorithm finds a set of actions and state variables that are sufficient for optimal planning, prior to any actual planning.

Our algorithm alternates between Process_Conditions and Find_Threats. Process_Conditions is responsible performing goal regression while checking whether each condition is implied by the start condition. Find_Threats is responsible for checking whether the conditions implied by the start condition, as found in Process_Conditions, are guaranteed to remain true throughout the trajectory. Our current implementation uses the theorem prover z3 to check whether the start condition implies each condition (De Moura and Bjørner 2008).

Algorithm 1 Both PROCESS_CONDITIONS and FIND_THREATS take in all defined variables as arguments.

procedure SCOPE_TASK($s_0, g, actions$)
  $discovered\_conds = g.split\_conjuncts()$
  $agenda = g.split\_conjuncts()$
  $used\_actions = []$
  while not agenda.empty() **do**
    $PROCESS\_CONDITIONS()$
    $FIND\_THREATS()$
  relevant_conds = discovered_conds - causal_links
  relevant_state_vars = $\bigcup_{c \in relevant\_conds} c.vars$
  return (relevant_state_vars, used_skills)
procedure PROCESS_CONDITIONS
  $cond = agenda.pop()$
  if $s_0 \implies cond$ **then**
    causal_links.push($cond$)
  else
    for $a \in actions$ **do**
      if $a.target\_vars \cap cond.vars \neq \emptyset$ **then**
        for $c \in a.prec.split\_disj()$ **do**
          if $c \notin discovered\_conds$ **then**
            $discovered\_conds.push(c)$
            $agenda$.push(c)
procedure FIND_THREATS
  $broken\_links = []$
  for $causal\_link \in causal\_links$ **do**
    for $action \in used\_actions$ **do**
      if $action.possibly\_effected\_vars \cap causal\_link.vars \neq \emptyset$ **then**
        $broken\_links.push(causal\_link)$
  for $causal\_link \in broken\_links$ **do**
    $causal\_links.remove(causal\_link)$

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 planning context, which allows for faster and more effective abstraction.

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

Helmert (2006) introduced Fast-Downward, a classical planning algorithm 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 determinized 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, their proposed method performs an exhaustive enumeration of all possible abstractions, which would be computationally intractable for large state-spaces.

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.

Boutilier, 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 Boutilier, 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.