Learning Probably Approximately Complete and Safe Action Models for Stochastic Worlds

Brendan Juba,[*1] **Roni Stern**[*2]

[1] Washington University in St. Louis
[2] Ben Gurion University & Xerox PARC
bjuba@wustl.edu, rstern@parc.com, sternron@post.bgu.ac.il

Abstract

We consider the problem of learning action models for planning in unknown stochastic environments that can be defined using the Probabilistic Planning Domain Description Language (PPDDL). As input, we are given a set of previously executed trajectories, and the main challenge is to learn an action model that has a similar goal achievement probability to the policies used to create these trajectories. To this end, we introduce a variant of PPDDL in which there is uncertainty about the transition probabilities, specified by an interval for each factor that contains the respective true transition probabilities. Then, we present SAM+, an algorithm that learns such an imprecise-PPDDL environment model. SAM+ has a polynomial time and sample complexity, and guarantees that with high probability, the true environment is indeed captured by the defined intervals. We prove that the action model SAM+ outputs has a goal achievement probability that is almost as good or better than that of the policies used to produced the training trajectories. Then, we show how to produce a PPDDL model based on this imprecise-PPDDL model that has similar properties.

Introduction

Domain-independent planning is a long-standing goal of Artificial Intelligence (AI) research. The input to a domain-independent planning algorithm traditionally includes a description of the domain in which we wish to plan. This domain description is usually specified in a formal language and includes an *action model*, which specifies which actions can be in a plan and how they work. In a simple planning languages such as STRIPS (Fikes and Nilsson 1971), this action model consists of the set of effects of the actions on the world state, and the set of preconditions that must hold in order for each action to be taken. Action models are notoriously hard to formally specify, even in such a simple planning language. This has motivated work on a number of methods for automatically learning these action models from examples (Yang, Wu, and Jiang 2007; Cresswell and Gregory 2011; Cresswell, McCluskey, and West 2013; Zhuo and Kambhampati 2013; Stern and Juba 2017; Aineto, Celorrio, and Onaindia 2019; Juba, Le, and Stern 2021).

[*]These authors contributed equally.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The challenge of specifying a domain model is only more acute in richer classes of action models. In this work, we consider domains in which the effects of actions are randomly determined each time an action is taken. Therefore, action models in such domains specify a distribution on effects for each action. Specifying such an action model by hand is extremely difficult: small errors in the probabilities may accumulate over the course of an execution, leading to wildly inaccurate estimates of the effects of a plan in the real world. It is therefore essential to use data about the real world to inform the model.

We follow an offline approach to permit safe learning. But, in order to establish that our model generalizes across goals, we consider a different learning paradigm. Following a recent approach to learning deterministic action models (Stern and Juba 2017; Juba, Le, and Stern 2021), we suppose that problems for a domain are sampled from a fixed distribution, and we are given a training set of trajectories executing policies aiming to solve those problems in the domain. We seek an action model that captures enough of the domain faithfully to ensure that

(i) policies that can be executed in the model behave similarly in the real world (solution safety) and
(ii) policies that attain similar rates of success on the problem distribution as the training policy distribution can be executed in the model (solution completeness).

Such a guarantee is similar to that provided by imitation learning (Osa et al. 2018; Khardon 1999), with the difference again (similar to the distinction with RL) that rather than seeking to match a human teacher's performance at a single objective, we would like our model to generalize across many possible goals.

We introduce an extension of PPDDL that captures a relaxed class of domain models in which there is uncertainty about the effects. Instead of giving a probability of an effect occurring, an interval is specified such that the actual probability lies somewhere in the interval—that is, specifying a MDP with Imprecise Probabilities (Satia and Lave Jr 1973). Then, we give an algorithm for learning such uncertain PPDDL models with the following guarantee: with high probability over both the training data and policy execution, the encountered transition distributions are consistent with the learned model. The learned model also satisfies an analogous completeness property, that ensures that there is a pol-

icy that the model guarantees to have a success rate that is similar to the policies used to generate the examples. Because probability bounds obtained in the model are guaranteed to hold for the real domain, such a model can be used for *safe* planning. Our model relies on having a transition distribution in which for a given pre-state, the fluents in the post-state are independent. This is in line with recent work on model-based RL. Without such an assumption, in general each action on each of the $2^{|F|}$ states yields a distribution over the $2^{|F|}$ possible states, where such a model requires $|A|2^{|F|}(2^{|F|}-1)$ parameters to specify,[1] which is clearly infeasible. Finally, we show how to create a model represented in Probabilistic Planning Domain Description Language (PPDDL) (Younes and Littman 2004), that has similar properties.

Background and Problem Definition

In this work, we assume the domain can be described using the Probabilistic Planning Domain Description (PPDDL) language. PPDDL is a formal language for specifying factored stochastic shortest path problems. We provide a brief desription of PPDDL below. For simplicity, we focus on the grounded version of PPDDL, and consider action costs to be unit for all actions.

A fluent f is a fact that may or may not be true in the domain. A state s in PPDDL is an assignment of values – true or false – to the set of fluents F in the domain, specifying which fluents are true in s. We refer to a state as a set, each element of which is an assignment to a single fluent. Let $s(f)$ be the value assigned to a fluent $f \in F$ in a state s. We refer to the assignments $s(f)=1$ and $s(f)=0$ (equiv., $s(\neg f)=1$) as literals. A partial state is an assignment of values to only a subset of the fluents in F. A partial state s' is consistent with a state s if the fluents specified by s' take the same values in s, i.e., $s' \subseteq s$. Let A be the set of actions that may be in a plan. An action model M for A specifies preconditions and effects for each action $a \in A$, denoted $\mathit{pre}_M(a)$ and $\mathit{eff}_M(a)$, respectively. The preconditions of an action is a partial state. An action a is applicable in a state s iff its preconditions are consistent with s. The effects of an action is a set of the form $\{\langle e_i, p_i\rangle\}_i$, where e_i is a partial state and p_i is the probability that it will *occur*. An effect $\langle e_i, p_i\rangle$ *occurring* means that after applying an action a at a state s we reach a state s' that is consistent with e_i even if e_i was not consistent with s. In this simple model, only the fluents in e_i may change from s to s'. We denote by $\Pr_M[s'|a,s]$ the probability, according to the action model M, that applying a in state s will result in reaching s'.

PPDDL supports advanced features such as recursive probabilistic effects and conditional effects, which we do not consider here. In addition, for most of this paper we limit the scope of our discussion to action models in which the effects are partial assignments of a set of literals whose distributions are independent. Formally, for every action a there is a set of literals E_a, and each literal $\ell \in E_a$ is associated with a marginal probability (factor) $\Pr[s'(\ell)|a, s(\neg\ell)]$. Every subset of literals $E'_a \subset E_a$ is an effect of a that occurs with probability

$$\prod_{\ell \in E'_a} \Pr[s'(\ell)|a, s(\neg\ell)] \cdot \prod_{\ell \in E_a \setminus E'_a} 1 - \Pr[s'(\ell)|a, s(\neg\ell)] \quad (1)$$

We refer to this assumption about effects as the *independent effects* assumption, and discuss how to relax it in the future work section .

A PPDDL planning *domain* is defined by a tuple $D = \langle F, A, M\rangle$ where F is the set of fluents, A is the set of actions, and M is the action model for these actions. A PPDDL planning *problem* is defined by a tuple $\langle D, s_I, G\rangle$ where D is a PPDDL domain; s_I is the start state, i.e., the state of the world before planning; and G is a partial state that define when a goal state has been found. A *solution* to a PPDDL problem $\Pi = \langle D, s_I, G\rangle$ is a *policy* $\pi : 2^F \to A$, mapping a state that may be encountered to an action. Executing a policy π on a problem Π means starting at s_I, applying the action $\pi(s_I)$, sampling a new state s', and continuing to apply actions according to π and the current state until some pre-defined stopping condition is met.[2] The resulting sequence of states and actions is called a *trajectory*.

Definition 1 (Trajectory)**.** *A trajectory is an alternating sequence of states and actions of the form* $(s_0, a_0, s_1, a_1, \ldots, a_n, s_n)$.

A single execution of a policy yields a trajectory where $s_0 = s_I$ and s_n is the last state reached in that execution. The length of a trajectory T, denoted $|T|$, is the number of actions in it. In the literature on learning action models (Wang 1994, 1995; Walsh and Littman 2008; Stern and Juba 2017; Arora et al. 2018, among others), it is common to represent a trajectory $T = \left\langle s_0, a_1, \ldots, a_{|T|}, s_{|T|}\right\rangle$ as a set of triples $\left\{ \langle s_{i-1}, a_i, s_i\rangle \right\}_{i=1}^{|T|}$. Each triple $\langle s_{i-1}, a_i, s_i\rangle$ is called an *action triplet*. The probability of observing T assuming an action model M, denoted $\Pr_{M,\pi}[T]$, is the product of the probabilities of T's constituent action triplets, i.e., $\Pr_{M,\pi}[T] = \prod_{(s,\pi(s),s') \in T} \Pr_M[s'|\pi(s), s]$.

To evaluate a policy for a given problem, we consider the set of trajectories that can be reached when following that policy, starting from the initial state. We denote this set of trajectories by $T(\pi, s)$, and denote by $T_G(\pi, s)$ its subset that includes only trajectories that reach the goal G. Common metrics include the expected number of actions in performed until a goal state is reached, and the *goal-achieving probability*. We mainly consider the latter option, denote by $\Pr_{M,\pi}[G|s]$ and given by $\sum_{T \in T_G(\pi,s)} \Pr_{M,\pi}[T]$.

Safe Planning Without an Action Model

We consider the setting where the planning agent is tasked to find a policy for a PPDDL problem $\Pi = \langle D, s_I, G\rangle$ but it has only a partial knowledge of the domain D. Specifically, the planner does not know the action model of the domain D. Instead, it is given a set of trajectories $\mathcal{T} = \{T_1, \ldots, T_m\}$ created by executing policies designed to solve problems

[1] Probability distributions must sum to 1.

[2] This stopping condition can be any property of the trajectory to that point, e.g., either reaching a goal state or performing a fixed number of actions, after which the agent gives up.

$\{\Pi_1, \ldots, \Pi_m\}$ in the same PPDDL domain, and in particular, all having the same action model.

Standard online RL setups approach this model-free setting by allowing the agent to perform exploratory actions, learning over time which policies are more effective than others. We focus on *offline learning*, where the objective is to learn an action model M' with which we can generate policies for a range of problems in the domain D. The main problem we address in this paper is to learn an action model that satisfies the following desirable properties:

1. **Safety.** Policies created using M' will also be applicable and effective for problems in the domain D.
2. **Completeness.** Policies that are effective in D will also be applicable and effective in the learned model M'.

We refer to the action model of D as the *real action model*, and denote the latter by M^*.

SAM Learning

The Safe Action Model (SAM) learning algorithm (Juba, Le, and Stern 2021; Stern and Juba 2017) has safety and completeness guarantees similar to those specified above. However, it is designed for *classical planning*, where states are fully observable and actions have deterministic effects. Classical planning is significantly simpler than our setting, where actions can have stochastic effects, Nevertheless, our work builds on SAM learning, so we describe it here briefly for completeness.

SAM learning, when applied to grounded domains, is based on the following simple rules, which apply for every observed action triplet $(s, a, s') \in T$ with $T \in \mathcal{T}$.

1. Rule 1 [not a precondition]. $\forall l \notin s : l \notin \mathit{pre}(a)$
2. Rule 2 [not an effect]. $\forall l \notin s' : l \notin \mathit{eff}(a)$
3. Rule 3 [must be an effect]. $\forall l \in s' \setminus s : l \in \mathit{eff}(a)$

where $\mathit{pre}(a)$ and $\mathit{eff}(a)$ are the preconditions and effects of a according to real action model M^*. We refer to these rules as the SAM learning rules. SAM learning works by initially assuming every action has all the literals as preconditions and none of the literals as effects, and then applying rules 1 and 3 above to remove preconditions and add effects as needed. In a classical planning setting where all actions are grounded, Juba et al. (2021) proved that the action model M_{SAM} created by SAM learning is: (1) *safe*, in the sense that every plan consistent with M_{SAM} is also consistent in the real action model, (2) *probably complete*, in the sense that with high probability, for most solvable problems there exists a plan that solves them and is consistent with M_{SAM}, given a number of trajectories that is polynomial in the number of fluents and actions. They also extended SAM learning to learn lifted action models and provided similar safety and completeness guarantees.

SAM+: SAM Learning for Stochastic Domains

We now introduce our method for learning action models for stochastic domains. It is an extension of SAM learning. The first SAM learning rule ("not a precondition") applies in our setting, since for every $\ell \in \mathit{pre}(a)$, we must have $\ell \in s$ for every s in which a is applied. The last SAM learning rule ("must be an effect") also applies to our setting, due to the standard frame axioms (i.e., state only changes due to actions of the agent). The second rule ("not an effect"), however, does not: a literal ℓ may be an effect of an action a even if there exists an action triplet (s, a, s') where $\ell \notin s'$. This may occur if ℓ is an effect of a but the probability of this effect occurring is not 1.0. This rule is crucial for SAM learning's safety guarantee, since it asserts that no unexpected effects will occur.[3] Thus, a strict form of safety as achieved for deterministic domains cannot be achieved in our setting. Instead, the SAM+ algorithm outputs an action model yielding a probabilistic form of safety, using the extension of PPDDL below, that we call *PPDDL with Imprecise Probabilities*.

PPDDL with Imprecise Probabilities

Markov Decision Processes with Imprecise Probabilities (MDP-IP) (Satia and Lave Jr 1973) are a representation of a *set* of MDPs. Instead of a transition function, an MDP-IP specifies a set of constraints over the transition function. In detail, an MDP-IP model M_{IP} defines a *transition credal set*, often denoted by $K_{M_{IP}}(s'|s, a)$, which maps state-action pairs to a set of constraints over the probability of reaching s' after performing action a in state s. M_{IP} represents every MDP M for which $\Pr_M[s'|s, a]$ is consistent with $K_{M_{IP}}(s'|s, a)$ for all triplets (s, a, s'). The MDP-IP representation captures the uncertainty about the real transition probabilities that arises when models are estimated from empirical observations.

Planners for MDP-IPs have been proposed, e.g., by Delgado, Sanner, and De Barros (2011). These planners seek policies that maximize some pessimistic objective—for example, for the reachability objective corresponding to a classical planning goal, we seek a policy that maximizes the minimum probability, over all MDPs consistent with the MDP-IP, of the policy reaching the goal. We will refer to this as the *maximin success probability*. An approximation to this objective that suffices for us assigns each distinct trajectory reaching the goal a "probability" equal to the products of the lower bounds of each transition. The sum of these values over the trajectories is a lower bound on the maximin success probability.

Inspired by MDP-IP, we propose the *PPDDL with Imprecise Probabilities* (PPDDL-IP) formalism. In a PPDDL-IP model, preconditions are defined identically to standard PPDDL models. But, each effect e in a PPDDL-IP model is defined with an interval $K_M[e|s, a] \subseteq [0, 1]$, specifying that the probability e occurs when performing a at state s is within that interval. For our independent effects assumption, we specify an interval for each factor ℓ, denoted $K_M[s'(\ell)|a, s(\neg\ell)]$.

Analogously to the relationship between MDP and MDP-IP, a PPDDL-IP domain specifies a set of PPDDL domains, for each assignment of probabilities to effects that satisfies the given interval constraints. We say that a PPDDL-IP action model M_{IP} is *safe* w.r.t. a PPDDL action model M if for every event e and possible action triplets (s, a, s'), $\Pr_M[e|s, a] \in K_{M_{IP}}(e|s, a)$. For example, if the credal sets are all $[0, 1]$, then M_{IP} is safe for any M (but useless).

[3] See proof by Stern et al.(2017).

The SAM+ Action Model

The SAM+ algorithm takes a set of trajectories $\mathcal{T}$ and a parameter $\delta > 0$, and outputs a PPDDL-IP action model denoted M_δ. We describe the preconditions and effects of M_δ below.

Preconditions. Let $\mathcal{T}(a)$ be all the action triplets for action a. States s and s' are said to be a *pre-* and *post-state* of a, respectively, if there is an action triplet $\langle s, a, s' \rangle \in \mathcal{T}(a)$. SAM+ sets the preconditions of an action a to be intersection over all the literals that were true in a pre-state of a.

$$pre_{M_\delta}(a) = \bigcap_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s \tag{2}$$

Effects. Note that we cannot distinguish whether or not ℓ was an effect of action a if $\ell \in s$, as it holds in s' in either case. We thus restrict attention to triplets where $\ell \notin s$ to estimate the credal set for ℓ: Let $\#_a(\ell \in s' \setminus s)$ and $\#_a(\ell \notin s)$ be the number of action triplets in $\mathcal{T}(a)$ in which ℓ is in the post-state and not the pre-state ($|\{\langle s, a, s' \rangle \in \mathcal{T}(a) : \ell \in s' \setminus s\}|$), and the number of action triplets in which ℓ was not in the pre-state ($|\{\langle s, a, s' \rangle \in \mathcal{T}(a) : \ell \notin s\}|$), respectively. SAM+ denotes the intervals $K_{M_\delta}[s'(\ell)|a, s(\ell)]$ by $K_\delta(s'(\ell)|a, s(\neg\ell))$, and computes them as follows.

1. If $\ell \in \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$ and $\ell \in \bigcap_{\langle s,a,s' \rangle \in \mathcal{T}(a):\ell \in s} s'$, then
$$K_\delta(s'(\ell)|a, s(\neg\ell)) = \left[1 - \frac{\ln(1/\delta)}{\#_a(\ell \notin s)}, 1\right] \tag{3}$$
2. If $\ell \in \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$ and $\ell \notin \bigcap_{\langle s,a,s' \rangle \in \mathcal{T}(a):\ell \in s} s'$, then
$$K_\delta(s'(\ell)|a, s(\neg\ell)) = \frac{\#_a(\ell \in s' \setminus s)}{\#_a(\ell \notin s)} \pm \sqrt{\frac{\ln(2/\delta)}{2\#_a(\ell \notin s)}} \tag{4}$$
3. If $\ell \notin \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$, then
$$K_\delta(s'(\ell)|a, s(\neg\ell)) = \left[0, \frac{\ln(1/\delta)}{\#_a(\ell \notin s)}\right] \tag{5}$$

If $\#_a(\ell \notin s) = 0$, then $K_\delta(s'(\ell)|a, s(\neg\ell)) = [0, 1]$. (In any case, we cap the credal sets at 0 and 1.) We remark that while the second interval is always valid, the first and third are smaller – i.e., more accurate – hence preferable for literals that appear deterministic.

An example of SAM+ executed on a simplified version of the Coffee domain of Dearden and Boutilier (1997) is available (Juba and Stern 2022). SAM+ has several attractive theoretical properties.

Theorem 1 (Efficiency)**.** *The SAM+ action model for m action triplets can be computed in time $O((m + |A|)|F|)$.*

Proof. We can compute the SAM+ action model in a single pass over the m triplets, taking time $O(m|F|)$: as in the deterministic SAM learning algorithms, we apply Rule 1 to each observed triplet to obtain the preconditions. We also compute counts $\#_a(\ell \in s' \setminus s)$ and $\#_a(\ell \notin s)$. These counts then may be used to produce the intervals given by Equations 4 and 5 for the effects in time $O(|A||F|)$. □

SAM+ is Probably Safe

The δ parameter is designed to represent the confidence that the learned model M_δ is correct w.r.t. the real action model M^*, i.e., that the range $K_\delta(s'(\ell)|a, s(\neg\ell))$ indeed includes the probability that ℓ will occur ($\Pr_{M^*}[s'(\ell)|a, s(\neg\ell)]$). Thus, increasing δ increases the range $K_\delta(s'(\ell)|a, s(\neg\ell))$. On the other hand, having more trajectories to learn from should yield a smaller range $K_\delta(s'(\ell)|a, s(\neg\ell))$ for a fixed δ. We now formalize this relation between $K_\delta(s'(\ell)|a, s(\neg\ell))$ and $\Pr[s'(\ell)|a, s(\ell)]$, and more generally M_δ and M^*.

Theorem 2 (Safety)**.** *For any $\delta' \geq 0$, any action applicable according to $M_{\delta'}$ is applicable in M^*. In addition, for $\delta' = \frac{\delta}{2|F||A|}$, $M_{\delta'}$ is correct for M^* with probability $1 - \delta$.*

Proof. The first part of Theorem 2 follows from prior work on SAM learning: since SAM+ assumes as preconditions a superset of the preconditions in the real action model, then if $\ell \in pre(a)$ for any action a, ℓ is also in a precondition for a in M_δ. Thus, if s satisfies the SAM+ precondition, s also satisfies $pre(a)$.

Next, we prove the second part of Theorem 2. Consider first any $\ell \in \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$. As trajectories are sampled from D, when a is taken in state s, $\ell \in s'$ with probability $\Pr_{M^*}[s'(\ell)|a, s]$. Consider the indicator random variables for $\ell \in s'$ for each of these events; note that if we subtract off their expectation (equal to $\Pr_{M^*}[s'(\ell)|a, s]$), the sum of these differences is a martingale sequence with differences bounded by 1, and the total number of such events in each trajectory, being determined by the policy in M^*, is a valid stopping time. So, for $\gamma = \sqrt{\frac{\ln(2/\delta)}{2|\{\langle s,a,s' \rangle \in \mathcal{T}(a):\ell \notin s\}|}}$ the Azuma-Hoeffding inequality (Azuma 1967; Hoeffding 1963) gives that for our $\#_a(\ell \notin s)$ such random variables, the fraction that take value 1, i.e., the empirical fraction $\frac{\#_a(\ell \in s' \setminus s)}{\#_a(\ell \notin s)}$ is within γ of its expectation, $\Pr_{M^*}[s'(\ell)|a, s]$, with probability at least $1 - 2e^{-2\gamma^2 \#_a(\ell \notin s)} = 1 - \delta'$. Thus, with probability $1 - \delta'$, $\Pr_{M^*}[s'(\ell)|a, s] \in K_{\delta'}(s'(\ell)|a, s(\neg\ell))$.

Similarly, for $\ell \notin \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$, we observe that if $\Pr_{M^*}[s'(\ell)|a, s] \geq \epsilon$ then since each s' is generated independently conditioned on s, the probability that $\ell \notin \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$ is at most $(1 - \epsilon)^{\#_a(\ell \notin s)}$. So, taking $\epsilon = \frac{\ln(1/\delta')}{\#_a(\ell \notin s)}$ we find that the probability is at most $(1 - \epsilon)^{\ln(1/\delta')/\epsilon} \leq e^{-\epsilon \ln(1/\delta')/\epsilon} = \delta'$.

The case for $\ell \in \bigcup_{\langle s,a,s' \rangle \in \mathcal{T}(a)} s' \setminus s$ such that $\ell \in \bigcap_{\langle s,a,s' \rangle \in \mathcal{T}(a):\ell \in s} s'$ is essentially identical: the probability that $\ell \in \bigcap_{\langle s,a,s' \rangle \in \mathcal{T}(a):\ell \in s} s'$ if $\Pr_{M^*}[\ell \notin s'|a, s] \geq \epsilon$ is at most $(1 - \epsilon)^{\#_a(\ell \notin s)}$. Thus again for $\epsilon = \frac{\ln(1/\delta')}{\#_a(\ell \notin s)}$, the probability of obtaining such an set of trajectories is at most δ'.

By a union bound over all $2|F|$ literals and $|A|$ actions, we find that all $\Pr_{M^*}[s'(\ell)|a, s] \in K_{\delta'}(s'(\ell)|a, s(\neg\ell))$ (so the action model is safe) with probability at least $1 - \delta$. □

SAM+ is Approximately Complete

Above we showed that the SAM+ action model can be learned efficiently and is safe w.r.t. to the real action model

M^* with high probability. Now, we show that it can be used to find policies that achieve the goal with high probability. To this end, we need to make some assumptions about how the given trajectories $\mathcal{T}$ were generated, and about the distribution of future problems we aim to solve in the domain.

Let $\mathcal{D}$ be a distribution over problems in the domain D, and let P be a (possibly probabilistic) planner that generates policies in this domain. We assume that the given set of trajectories $\mathcal{T}$ was created by sampling a planning problem Π from $\mathcal{D}$ and executing a policy π created by P for this problem. We say that an action model is *approximately complete* if the policy used in the sampling distribution achieves the goal of the sampled problem with probability p, then the SAM+ model produces policies that solve the sampled problems with probability at least $p - \epsilon$ in D. We stress that this probability is both over the stochastic transitions of the MDP and the sampling of a problem from $\mathcal{D}$: for example, a $\mathcal{D}$ for which the goal cannot be achieved with probability $1/2$, and otherwise a policy succeeds with probability $2p$, yields an overall probability p of solving the problem.

Theorem 3 (Approximate completeness)**.** *Fix a distribution on policies such that for the distribution $\mathcal{D}$ over problems in the PPDDL domain D, the policies solve the problems with probability p and runs for L steps in expectation, the draw from $\mathcal{D}$, and draw of the policy. Given $m \geq \frac{4096|A|^2|F|^3L^2}{(1-\epsilon)^4\epsilon^4} \ln \frac{4|F||A|}{\delta}$ trajectories independently drawn from the policies for problems from $\mathcal{D}$, with probability $1 - 2\delta$, the expected maximin success probability in the SAM+ action model, over problems Π sampled from $\mathcal{D}$, is at least $p - \epsilon$. In particular, if we execute a policy that has a maximum lower bound of solving Π in the SAM+ model, Π is solved with probability at least $p - \epsilon$ over both the draw of Π and execution in D.*

To prove Theorem 3, we first observe that we can afford to ignore actions that are seldom used in the examples to solve problems drawn from $\mathcal{D}$; that is, we can restrict our attention to *useful* actions in the following sense:

Definition 2. *An action a is said to be* ϵ-useful *with respect to a domain D, a planner P, and a distribution on problems $\mathcal{D}$, if with probability at least $1 - \epsilon$, a appears in a trajectory sampled by executing in D a policy obtained by giving P a problem drawn from $\mathcal{D}$.*

Indeed, with high probability, problems drawn from $\mathcal{D}$ can be solved by policies that only use useful actions.

Lemma 1. *Suppose trajectories sampled as in Definition 2 solve the sampled problem Π with probability p. Then problems sampled from $\mathcal{D}$ can be solved by policies that only use $\epsilon/|A|$-useful actions with probability at least $p - \epsilon$.*

Proof. Suppose we sample Π from $\mathcal{D}$ and run the planner on Π to obtain a policy π. Now, we consider the policy $\tilde{\pi}$ obtained by modifying π by replacing all actions that are not $\epsilon/|A|$-useful with termination. We observe that conditioned on π only producing useful actions, $\tilde{\pi}$ is identical to π. By a union bound over the inadequate actions, the probability that π uses any action that is not $\epsilon/|A|$-useful is at most ϵ. Thus, $\tilde{\pi}$ solves the random problem Π with probability at least $p - \epsilon$, and only uses $\epsilon/|A|$-useful actions, as needed. □

Moreover, we can guarantee that we will observe each of the useful actions many times:

Lemma 2. *Among $m \geq 8|A| \ln \frac{2|A|}{\delta} \frac{1}{\epsilon}$ trajectories, each $\epsilon/|A|$-useful action occurs in at least $m' \geq \frac{\epsilon}{2|A|}m$ of the trajectories with probability at least $1 - \delta/2$*

Proof. Consider any $\epsilon/|A|$-useful action a. For each of the m trajectories, we define an indicator random variable indicating whether or not a is used in that trajectory. Since a is $\epsilon/|A|$-useful, these random variables all have expected value at least $\epsilon/|A|$. Since each of the trajectories are sampled independently, these random variables are mutually independent, and by a Chernoff bound, the probability that a appears in fewer than $\frac{\epsilon}{2|A|}m$ of the trajectories is at most $e^{-\frac{1}{2\cdot 2^2}m\frac{\epsilon}{|A|}} \leq \frac{\delta}{2|A|}$ Now, taking a union bound over the $\epsilon/|A|$-useful actions, we find that they all appear at least $\frac{\epsilon}{2|A|}m$ times with probability at least $1-\delta/2$, as needed. □

In particular, the widths of the intervals $K_\delta[s'(\ell)|a, s(\neg\ell)]$ in the SAM+ model shrink as we observe the actions more frequently; this enables us to ensure that the SAM+ model is *adequate* for accurate planning in the following sense:

Definition 3. *We say that the the SAM+ action model M_δ is* (ϵ_1, ϵ_2)-adequate *for action a if with probability $1 - \epsilon_1$ over trajectories sampled as in Definition 2, when a is used in a state s of the trajectory,*

1. *the preconditions of a in M_δ are satisfied for s*
2. *for all ℓ the width of the interval $K_{\delta'}(s'(\ell)|a, s(\neg\ell))$ is at most ϵ_2.*

Lemma 3. *If $\mathcal{T}(a)$ contains at least $m' \geq \frac{16|A||F|}{\epsilon_1\epsilon_2^2} \ln \frac{2}{\delta'}$ triplets, then the SAM+ action model for a is $(\frac{\epsilon_1}{|A|}, \epsilon_2)$-adequate with probability $1 - \delta/2|A|$.*

Proof. For the first condition, observe that if a literal ℓ is not in *pre*(a) and occurs in some state s of a sampled trajectory with probability greater than $\frac{\epsilon_1}{4|A||F|}$, then the probability that ℓ is in the SAM+ precondition for a is at most $(1 - \frac{\epsilon_1}{4|A||F|})^{m'} \leq \frac{\delta}{8|A||F|}$ Thus, taking a union bound over all $2|F|$ literals, we find that with probability $1 - \delta/8|A|$, no such literals are present in the SAM+ preconditions of a (i.e., in *pre*$_{M_\delta}(a)$). Now, moreover, also by a union bound over the literals not in *pre*(a), the probability that any of the literals ℓ not in *pre*(a) that occur in some state s of a sampled trajectory with probability greater than $\frac{\epsilon_1}{4|A||F|}$ actually occur in a state s where a is used in a sampled trajectory is at most $\epsilon_1/2|A|$.

For the second condition, we observe that for each literal ℓ, the width of the interval for a is at most $\sqrt{\frac{2\ln(2/\delta')}{|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}|}}$. In particular, consider any ℓ that is is not in *pre*(a) such that ℓ does not occur in some state s of a sampled trajectory with probability greater than $\frac{\epsilon_1}{4|A||F|}$. If we consider the indicator random variable for whether a sampled trajectory contains a triplet in which a is taken in some state s such that $\ell \notin s$, we see that these are independent random variables with expected value at least $\frac{\epsilon_1}{4|A||F|}$.

So, by a Chernoff bound, with probability $1-\frac{\delta}{8|A||F|}$, at least $\frac{\epsilon_1}{8|A||F|}m' = \frac{2}{\epsilon_2^2}\ln\frac{2}{\delta'}$ out of the m' triplets have s such that $\ell \notin s$. Hence, the width of the interval $K_\delta[s'(\ell)|a, s(\neg\ell)]$ is at most $\sqrt{\frac{2\ln(2/\delta')}{2\ln(2/\delta')/\epsilon_2^2}} = \epsilon_2$ as needed. In turn, by a union bound over the literals that occur with probability less than $\frac{\epsilon_1}{4|A||F|}$ when a is taken, the probability that any of these occur in any state of a sampled trajectory where a is taken is at most $\frac{\epsilon_1}{2|A|}$. Thus, with probability $1-\frac{\epsilon_1}{2|A|}$, all of the intervals have width at most ϵ_2.

By union bounds over the two cases, with probability $1-\frac{\delta}{2|A|}$, we obtain a SAM+ action model such that for a sampled trajectory, each condition holds with probability $\frac{\epsilon_1}{2|A|}$; by another union bound over the conditions, both simultaneously hold in the trajectory with probability $\frac{\epsilon_1}{|A|}$. □

Adequate action models include solutions to problems from $\mathcal{D}$ with success rates similar to the training distribution:

Lemma 4. *Suppose that the SAM+ action model is $(\frac{\epsilon_1}{|A|}, \epsilon_2)$-adequate for all $\frac{\epsilon_3}{|A|}$-useful actions. Then with probability $1-\delta$, if the sampling distribution solves the problems from $\mathcal{D}$ with probability p while taking L steps in expectation, the SAM+ action model yields an expected maximin success probability of at least $p-(\epsilon_1+\frac{\epsilon_2\cdot L\cdot|F|}{(1-\epsilon_1)(1-\epsilon_3)}+\epsilon_3)$. Moreover, there exist policies attaining this probability that only take actions for which the width of the intervals for all literals is at most ϵ_2 for the states in which they are invoked.*

Proof. We start by extending Lemma 1: for the policies π produced by the planner for problems sampled from D, we consider $\tilde{\pi}$ that executes π until it would either take an action that is not $\frac{\epsilon_3}{|A|}$-useful, or would invoke an action in a state for which some literal has a credal set of width greater than ϵ_2, and terminates in either of these conditions. We observe that $\tilde{\pi}$ has trajectories that are no longer than π in any sample. Moreover, by a union bound over the actions that are not $\frac{\epsilon_3}{|A|}$-useful, the original distribution over trajectories only produces a trajectory that uses any of these actions with probability at most ϵ_3. Since by hypothesis, the action model is $(\frac{\epsilon_1}{|A|}, \epsilon_2)$-adequate for all the useful actions, the probability that π would take one of these actions in a state where the credal sets are wider than ϵ_2 for any literal is at most $\frac{\epsilon_1}{|A|}$; by a union bound over the useful actions, we find thus that with probability $1-(\epsilon_1+\epsilon_3)$, neither of these events occurs and π produces an execution that is identical to that produced by $\tilde{\pi}$. In particular, for a maximin π, $\tilde{\pi}$ must also solve the problem Π sampled from $\mathcal{D}$ with probability at least $p-(\epsilon_1+\epsilon_3)$.

Since by construction, $\tilde{\pi}$ only takes actions for which the width of the interval is at most ϵ_2, we can bound the expected maximin success probability $\tilde{p}$ of the SAM+ action model using $\tilde{\pi}$ as follows.

$$\begin{aligned}
\tilde{p} &\geq \sum_{\Pi,\pi} \Pr_{D,P}[\Pi,\pi] \sum_{\substack{\text{trajectories } T\\ \text{of } \tilde{\pi} \text{ solving } \Pi}} \Pr_{\tilde{\pi},M}[T|\Pi] \\
&= \sum_{\Pi,\pi} \Pr_{D,P}[\Pi,\pi] \sum_{k=0}^{\infty} \sum_{\substack{\text{trajectories } T\\ \text{of } \tilde{\pi} \text{ solving}\\ \Pi \text{ in } k \text{ steps}}} \prod_{i=1}^{k} \Pr_{\tilde{\pi},M}[T_i|T_{i-1}] \\
&\geq \sum_{\Pi,\pi} \Pr_{D,P}[\Pi,\pi] \sum_{k=0}^{\infty} \sum_{\substack{\text{trajectories } T\\ \text{of } \tilde{\pi} \text{ solving}\\ \Pi \text{ in } k \text{ steps}}} (1-k\epsilon_2|F|) \Pr_{\tilde{\pi},M^*}[T|\Pi] \\
&\geq p - \frac{L}{(1-\epsilon_1)(1-\epsilon_3)}\epsilon_2|F| - (\epsilon_1+\epsilon_3)
\end{aligned}$$

Since by Theorem 2 the SAM+ action model is safe with probability $1-\delta$, the policies obtained from executing the planner on the SAM+ action model indeed obtain a probability of success that is at least the expected lower bound on $\tilde{\pi}$, $p-(\epsilon_1+\frac{\epsilon_2\cdot L\cdot|F|}{(1-\epsilon_1)(1-\epsilon_3)}+\epsilon_3)$ with probability $1-\delta$. □

By the above reasoning, we obtain Theorem 3. Formally:

Proof of Theorem 3. We first note that the quoted number of actions is sufficient to obtain that by Lemma 2, for $\epsilon_3 = \epsilon/4$, each of the $\epsilon_3/|A|$-useful actions occurs in at least $m' \geq \frac{512|A||F|^2L^2}{(1-\epsilon)^2\epsilon^3}\log\frac{4|F||A|}{\delta}$ of the traces with probability at least $1-\delta/2$. In turn, by Lemma 3, for each of these actions, for $\epsilon_1 = \epsilon/4$ and $\epsilon_2 = \frac{\epsilon(1-\epsilon)^2}{2|F|L}$, the $\epsilon_3/|A|$-useful actions have $(\epsilon_1/|A|, \epsilon_2)$-adequate action models with probability $1-\delta/2|A|$ each; a union bound over these actions gives that overall, they are simultaneously adequate with probability $1-\delta/2$, and hence with probability $1-\delta$ overall, we have an adequate action model for all of the $\epsilon_3/|A|$-useful actions. So finally, by Lemma 4, with probability $1-2\delta$ overall, problems drawn from D have an expected maximin success probability at least $p-\epsilon$ under the SAM+ action model, as claimed. □

SAM+ with PPDDL Planners

Actually, Lemma 4 gives us slightly more than needed for Theorem 3: the policy only needs to consider actions for which the intervals $K_\delta[s'(\ell)|a,s]$ are narrow. So, we can take the midpoint of these intervals as an estimate of the probability in a standard PPDDL encoding, if we include preconditions that prevent a solution from executing actions that would lead to intervals that are too wide for any of the factors. In more detail: First, let us suppose L' is an *upper bound* on the lengths of plans we consider. For $\ell \in \bigcup_{\langle s,a,s'\rangle\in\mathcal{T}(a)} s' \setminus s$ with $\ell \notin \bigcap_{\langle s,a,s'\rangle\in\mathcal{T}(a):\ell\in s} s'$, $\ell \in \mathit{pre}(a)$ iff

$$|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}| < \frac{8|F|^2L'^2}{(1-\epsilon)^4\epsilon^2}\ln\frac{4|F||A|}{\delta},$$

and otherwise ($\ell \notin \bigcup_{\langle s,a,s'\rangle \in \mathcal{T}(a)} s' \setminus s$ or $\ell \in \bigcap_{\langle s,a,s'\rangle \in \mathcal{T}(a):\ell \in s} s'$), $\ell \in \mathit{pre}(a)$ iff

$$|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}| < \frac{2|F|L'}{\epsilon(1-\epsilon)^2} \ln \frac{2|F||A|}{\delta}.$$

We note that the original preconditions correspond to the set of literals ℓ such that $|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}| = 0$, so this is a superset, i.e., a stricter precondition. Moreover, for these literals, we indeed have respectively that the widths of the confidence intervals for literals *not* included in the precondition are at most $\frac{\epsilon(1-\epsilon)^2}{2|F|L'} = \epsilon_2$.

Now, instead of the intervals, we use the following factors for the transition probabilities: For $\ell \in \bigcup_{\langle s,a,s'\rangle \in \mathcal{T}(a)} s' \setminus s$ with $\ell \notin \bigcap_{\langle s,a,s'\rangle \in \mathcal{T}(a):\ell \in s} s'$, the transition probability factor for ℓ given $\ell \notin s$ and a is an empirical estimate of the probability:

$$\Pr[s'(\ell)|a, s(\neg\ell)] = \frac{|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \in s' \setminus s\}|}{|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}|} \quad (6)$$

for $\ell \in \bigcup_{\langle s,a,s'\rangle \in \mathcal{T}(a)} s' \setminus s$ with $\ell \in \bigcap_{\langle s,a,s'\rangle \in \mathcal{T}(a):\ell \in s} s'$ it is

$$\Pr[s'(\ell)|a, s(\neg\ell)] = 1 - \frac{\ln(2|F||A|/\delta)}{2|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}|} \quad (7)$$

and otherwise (i.e., for $\ell \notin \bigcup_{\langle s,a,s'\rangle \in \mathcal{T}(a)} s' \setminus s$),

$$\Pr[s'(\ell)|a, s(\neg\ell)] = \frac{\ln(2|F||A|/\delta)}{2|\{\langle s,a,s'\rangle \in \mathcal{T}(a) : \ell \notin s\}|} \quad (8)$$

i.e., the midpoints of our previous intervals.

Whereas the intervals contained the true transition probabilities with high probability, we now only have that our point estimates of these transition probabilities are $\epsilon_2/2$-close to the true probabilities on each transition. We can thus (only) guarantee an approximate form of safety:

Theorem 4 (Approximate safety—PPDDL)**.** *With probability $1-\delta$, the SAM+ PPDDL action model satisfies the following: the probability any plan of length at most L' succeeds in the SAM+ PPDDL model is at most $(1+\epsilon)$ times greater than under the true model M^*. In particular, all actions that are applicable in a plan under the SAM+ PPDDL model are applicable in M^*.*

Proof. We recall that by Theorem 2, the PDDL-IP model created by SAM+ is correct w.r.t M^*. The midpoints of its intervals are, by construction, at most $\epsilon_2/2$-far from the true probability for each factor. Thus, our estimated probability of success $\hat{p}$ for the policy π for the problem Π is at most

$$\begin{aligned}
\hat{p} &\leq \sum_{k=0}^{L'} \sum_{\substack{\text{trajectories } T \\ \text{of } \pi \text{ solving} \\ \Pi \text{ in } k \text{ steps}}} \prod_{i=1}^{k} \Pr_{\pi,M}[T_i|T_{i-1}] \\
&\leq \sum_{k=0}^{L'} \sum_{\substack{\text{trajectories } T \\ \text{of } \pi \text{ solving} \\ \Pi \text{ in } k \text{ steps}}} (1+|F|\epsilon_2/2)^k \prod_{i=1}^{k} \Pr_{\pi,M^*}[T_i|T_{i-1}] \\
&\leq \sum_{k=0}^{L'} \sum_{\substack{\text{trajectories } T \\ \text{of } \pi \text{ solving} \\ \Pi \text{ in } k \text{ steps}}} (1+k|F|\epsilon_2/2 + \left(\frac{k|F|\epsilon_2}{2}\right)^2) \Pr_{\pi,M^*}[T|\Pi] \\
&\leq (1+\epsilon)p
\end{aligned}$$

where the second to last line used $1+x \leq e^x \leq 1+x+x^2$ for all $x \leq 1$; note here that we have chosen $\epsilon_2 = \frac{\epsilon}{L'|F|}$ so that $k|F|\epsilon_2 \leq \epsilon < 1$ for all $k \leq L'$, so we can invoke this inequality, and $(\epsilon/2)^2 < \epsilon/2$. The moreover part follows directly from Theorem 2 and the fact that the PPDDL preconditions are only stronger than the original. □

In turn, we obtain the following guarantee for completeness, essentially analogously to the original argument:

Theorem 5 (Approximate completeness—PPDDL)**.** *Fix a planner, and suppose that for the distribution $\mathcal{D}$ over problems in a domain D, the planner produces a policy that solves the problems with probability p and runs for L steps in expectation, the draw from $\mathcal{D}$, and the planner itself. Given $m \geq \frac{4096|A|^2|F|^3L^2}{(1-\epsilon)^4\epsilon^4} \ln \frac{4|F||A|}{\delta}$ trajectories independently drawn from the planner on problems from $\mathcal{D}$, with probability $1-2\delta$, the SAM+ action model satisfies the following: when a problem Π is sampled from $\mathcal{D}$ and we execute a policy of length at most $\frac{1}{\epsilon}L$ that maximizes the probability of solving Π in the SAM+ PPDDL model with $L' = \frac{1}{\epsilon}L$, Π is solved with probability at least $p-3\epsilon$ (over both the draw of Π and execution in M^*).*

Proof. We first note that by Markov's inequality, that if $\tilde{\pi}$ runs for L steps in expectation, then with probability $1-\epsilon$ it runs for at most $\frac{1}{\epsilon}L$ steps. Theorem 4 guarantees that p is overestimated by at most ϵ on all policies that execute for at most $\frac{1}{\epsilon}L$ steps. In turn, the success probability of an optimal policy is underestimated by at most ϵ, following the original proof of Theorem 3. All together, we find that with probability $p-\epsilon$, the optimal policy solves the problem Π in at most $\frac{1}{\epsilon}L$ steps, and achieves an estimated probability under M that is at most ϵ smaller. Hence, the optimal policy under M that overestimates by at most ϵ is succeeding with probability at least $p-3\epsilon$, as claimed. □

Related Work

There has been work on learning noisy STRIPS operators from incomplete observations (Pasula, Zettlemoyer, and Kaelbling 2007; Mourao et al. 2012; Rodrigues, Gerard, and Rouveirol 2011; Ng and Petrick 2019). They propose

methods for learning PPDDL representations from incomplete observations, a more demanding setting than ours. But, they do not provide any theoretical guarantees. Similarly, Martínez et al. (2017) consider learning richer probabilistic domain models that also incorporate exogenous effects; but, they do not have time or sample complexity guarantees, nor do they connect the objective they optimize to the quality of the model.

This problem of learning an action model is related to Reinforcement Learning (RL), with the crucial difference that RL typically assumes that we only wish to design a policy for a single, fixed, learned reward function, whereas here we wish to use the same domain model for various goals given as input. Nevertheless, the central challenges limiting the state-of-the-art in RL apply here as well. In particular, we would like methods for learning domain models with guarantees of efficiency and correctness. Ideally, we would of course like an algorithm that finds a domain model that is close to the true domain and, if states are described by a set of (Boolean) fluents $|F|$, has both time complexity and sample complexity bounded by a polynomial in $|F|$. Unfortunately, in general this is too much to hope for, since there are $2^{|F|}$ states. Concretely, along the lines of Kakade (2003), domains that capture a simple "combination lock" require $\Omega(2^{|F|}|A|)$ trials to solve, if the combination is described by the fluents and A is the set of actions. Consequently, algorithms with theoretical guarantees for learning optimal policies in a general RL setting seek to observe every action in every state, and this is optimal (Strehl et al. 2006). This may be viewed as a difficulty with "exploring" the environment (the state with the open lock is hard to reach) or a difficulty with "generalizing" across states (an action suddenly opens the lock in the state where the correct combination is entered). The methods for circumventing such problems in RL frequently involve assuming some kind of linear structure on the reward or value function (Osband and Van Roy 2014; Osband, Roy, and Wen 2016; Jin et al. 2020; Yang and Wang 2019), which prevents them from capturing STRIPS-style goals, which are given by a possibly large conjunction of the fluent settings. By contrast, work on Model-Based RL (MBRL) (Kearns and Koller 1999; Koller and Parr 2000; Strehl, Diuk, and Littman 2007; Diuk, Li, and Leffler 2009), attempts to fit the environment dynamics to a specific model class; classically, these were based on Dynamic Bayesian Network models (Dean and Kanazawa 1989) of the transition distribution, which may be much more compact, by assuming that the various factors are generated independently.

A problem with even MBRL algorithms – and in particular, for the aforementioned methods for MBRL – is that they still incentivize an agent to visit unexplored portions of the state space (w.r.t. the conditional probability tables), which may be *unsafe*. This has posed an obstacle to the adoption of RL in practice; in turn, it has motivated recent work on *offline RL* (Levine et al. 2020; Kidambi et al. 2020; Yu et al. 2020), which seeks to learn policies, in particular *safe* policies, using a training set of trajectories collected in the domain. These trajectories might, for example, be provided by a human demonstrating good behavior. Safety here generally means that we guarantee with high probability that the policy will remain in some set of "safe" states (Thomas 2015; Thomas et al. 2019). These methods are cast in a traditional RL, discounted-reward setting, and simply charge a penalty for uncertainty, which is not a good match to the reachability goals we consider. Moreover, how and when can we hope to guarantee that the model transfers to other goals?

Conclusion and Future Work

We have shown how to safely and efficiently acquire models of stochastic domains that nevertheless generalize to goals beyond those for which solutions were previously demonstrated. Supposing that we have an agent with a fixed set of actions and we have determined a set of fluents that the agent can observe, we collect a set of trajectories demonstrating how to solve problems in the domain—for example, these may be collected by a human operator, or by an inefficient algorithm. The demonstrations need not always achieve the goal, and indeed need not follow a fixed or known policy. When we provide these trajectories to SAM+, it produces an action model in which the optimal policy's success probability at least matches the success rate of the demonstrations.

Although the style of domain model we used here is a relatively limited fragment of PPDDL that uses propositional fluents and assumes all fluents transition independently, the technique is not inherently limited to such models. It is straightforward to extend this algorithm using the approach of Juba, Le, and Stern (2021) to learn lifted domain models, at least when the "injective binding assumption" holds, i.e., when two parameters of the same type are not bound to the same object in the example trajectories. We believe it should be possible to extend this further to handle the kind of "deictic" references considered by Pasula, Zettlemoyer, and Kaelbling (2007). It is also straightforward to extend to domains in which the transitions depend on a constant-size set of attributes in the previous state, using the technique of Strehl, Diuk, and Littman (2007). This captures a family of conditional effects, in which any given fluent only has conditional effects depending on a small family of other fluents. Or, indeed, it may similarly be extended to domains in which the fluents take values from a larger discrete set. In turn, this allows us to consider domains in which there are dependencies among small sets of fluents, by treating the entire block as a vector-valued fluent.

There are, of course, limits to how far we may extend the approach: learning arbitrary graphical model representations with unknown structures is known to be hard in various senses (Chickering 1996; Chickering, Heckerman, and Meek 2004). But, concretely, it is still not clear how rich a fragment of PPDDL we can learn efficiently: Can we learn products of arbitrary small-support distributions? Can we learn exogenous effects? What about models of domains with continuous state spaces? We leave these to future work.

Acknowledgements

We thank our reviewers for their constructive comments. This research is partially funded by NSF awards IIS-1908287 and CCF-1718380, and BSF grant #2018684 to

Roni Stern, and by the Defense Advanced Research Projects Agency (DARPA) as part of the SAIL-ON program.

References

Aineto, D.; Celorrio, S.; and Onaindia, E. 2019. Learning action models with minimal observability. *Artificial Intelligence*, 275: 104–137.

Arora, A.; Fiorino, H.; Pellier, D.; Etivier, M.; and Pesty, S. 2018. A review of learning planning action models. *Knowledge Engineering Review*, 33.

Azuma, K. 1967. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3): 357–367.

Chickering, D. M. 1996. Learning Bayesian networks is NP-complete. In *Learning From Data*, 121–130. Springer.

Chickering, M.; Heckerman, D.; and Meek, C. 2004. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5: 1287–1330.

Cresswell, S.; and Gregory, P. 2011. Generalised domain model acquisition from action traces. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 42–49.

Cresswell, S. N.; McCluskey, T. L.; and West, M. M. 2013. Acquiring planning domain models using LOCM. *The Knowledge Engineering Review*, 28(2): 195–213.

Dean, T.; and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2): 142–150.

Dearden, R.; and Boutilier, C. 1997. Abstraction and approximate decision-theoretic planning. *Artificial Intelligence*, 89(1-2): 219–283.

Delgado, K. V.; Sanner, S.; and De Barros, L. N. 2011. Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence*, 175(9-10): 1498–1527.

Diuk, C.; Li, L.; and Leffler, B. R. 2009. The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning. In *International Conference on Machine Learning (ICML)*, 249–256.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4): 189–208.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58: 13–30.

Jin, C.; Yang, Z.; Wang, Z.; and Jordan, M. I. 2020. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory (COLT)*, 2137–2143.

Juba, B.; Le, H. S.; and Stern, R. 2021. Safe Learning of Lifted Action Models. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 379–389.

Juba, B.; and Stern, R. 2022. An Example of the SAM+ Algorithm for Learning Action Models for Stochastic Worlds. https://arxiv.org/abs/2203.12499. Accessed: 2022-04-11.

Kakade, S. M. 2003. *On the Sample Complexity of Reinforcement Learning*. Ph.D. thesis, University College London.

Kearns, M.; and Koller, D. 1999. Efficient reinforcement learning in factored MDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 740–747.

Khardon, R. 1999. Learning to take actions. *Machine Learning*, 35(1): 57–90.

Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. MOReL: Model-Based Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems 33*.

Koller, D.; and Parr, R. 2000. Policy iteration for factored MDPs. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 326–334.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. Technical report.

Martínez, D.; Alenya, G.; Ribeiro, T.; Inoue, K.; and Torras, C. 2017. Relational reinforcement learning for planning with exogenous effects. *Journal of Machine Learning Research*, 18(78): 1–44.

Mourao, K.; Zettlemoyer, L.; Petrick, R.; and Steedman, M. 2012. Learning STRIPS Operators from Noisy and Incomplete Observations. In *Proceedings of the Twenty Eighth Conference on Uncertainty in Artificial Intelligence (UAI 2012)*, 614–623.

Ng, J. H. A.; and Petrick, R. P. 2019. Incremental Learning of Planning Actions in Model-Based Reinforcement Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3195–3201.

Osa, T.; Pajarinen, J.; Neumann, G.; Bagnell, J. A.; Abbeel, P.; Peters, J.; et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2): 1–179.

Osband, I.; Roy, B. V.; and Wen, Z. 2016. Generalization and Exploration via Randomized Value Functions. In *International Conference on Machine Learning (ICML)*, 2377–2386.

Osband, I.; and Van Roy, B. 2014. Near-optimal Reinforcement Learning in Factored MDPs. In *Advances in Neural Information Processing Systems 27*, 604–612.

Pasula, H. M.; Zettlemoyer, L. S.; and Kaelbling, L. P. 2007. Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, 29: 309–352.

Rodrigues, C.; Gerard, P.; and Rouveirol, C. 2011. Incremental Learning of Relational Action Models in Noisy Environments. In *International Conference on Inductive Logic Programming (ILP)*, 206–213.

Satia, J. K.; and Lave Jr, R. E. 1973. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3): 728–740.

Stern, R.; and Juba, B. 2017. Efficient, Safe, and Probably Approximately Complete Learning of Action Models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 4405–4411.

Strehl, A. L.; Diuk, C.; and Littman, M. L. 2007. Efficient structure learning in factored-state MDPs. In *AAAI Conference on Artificial Intelligence (AAAI)*, 645–650.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. PAC model-free reinforcement learning. In *International Conference on Machine Learning (ICML)*, 881–888.

Thomas, P. S. 2015. *Safe reinforcement learning*. Ph.D. thesis, University of Massachusetts, Amherst.

Thomas, P. S.; da Silva, B. C.; Barto, A. G.; Giguere, S.; Brun, Y.; and Brunskill, E. 2019. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468): 999–1004.

Walsh, T. J.; and Littman, M. L. 2008. Efficient learning of action schemas and web-service descriptions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 714–719.

Wang, X. 1994. Learning planning operators by observation and practice. In *Second International Conference on Artificial Intelligence Planning Systems (AIPS)*, 335–340.

Wang, X. 1995. Learning by observation and practice: an incremental approach for planning operator acquisition. In *International Conference on Machine Learning (ICML)*, 549–557.

Yang, L. F.; and Wang, M. 2019. Sample-Optimal Parametric Q-Learning Using Linearly Additive Features. In *International Conference on Machine Learning (ICML)*, 6995–7004.

Yang, Q.; Wu, K.; and Jiang, Y. 2007. Learning action models from plan examples using weighted MAX-SAT. *Artificial Intelligence*, 171(2-3): 107–143.

Younes, H. L.; and Littman, M. L. 2004. PPDDL1. 0: An extension to PDDL for expressing planning domains with probabilistic effects. *Technical Report CMU-CS-04-162*.

Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. MOPO: Model-based Offline Policy Optimization. In *Advances in Neural Information Processing Systems 33*, 14129–14142.

Zhuo, H. H.; and Kambhampati, S. 2013. Action-model acquisition from noisy plan traces. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2444–2450.