CORPP: Commonsense Reasoning and Probabilistic Planning, as Applied to Dialog with a Mobile Robot

Shiqi Zhang and **Peter Stone**
Department of Computer Science, The University of Texas at Austin, Austin, TX, USA
{szhang,pstone}@cs.utexas.edu

Abstract

In order to be fully robust and responsive to a dynamically changing real-world environment, intelligent robots will need to engage in a variety of simultaneous reasoning modalities. In particular, in this paper we consider their needs to i) reason with commonsense knowledge, ii) model their nondeterministic action outcomes and partial observability, and iii) plan toward maximizing long-term rewards. Answer Set Programming (ASP) is good at representing and reasoning with commonsense and default knowledge, and Partially Observable Markov Decision Processes (POMDPs) are strong at planning under uncertainty toward maximizing long-term rewards. This paper introduces the CORPP algorithm which combines P-log, a probabilistic extension of ASP, with POMDPs to integrate commonsense reasoning with planning under uncertainty. Our approach is fully implemented and tested on a shopping request identification problem both in simulation and on a real robot. Compared with existing approaches using P-log or POMDPs individually, we observe significant improvements in both efficiency and accuracy.

1 Introduction

Intelligent robots are becoming increasingly useful across a wide range of tasks. In real-world environments, intelligent robots need to be capable of representing and reasoning with logical and probabilistic commonsense knowledge. Additionally, due to the fundamental dynamism of the real world, intelligent robots have to be able to model and reason about quantitative uncertainties from nondeterministic action outcomes and unreliable local observations. While there are existing methods for dealing separately with either reasoning with commonsense knowledge or planning under uncertainty, to the best of our knowledge, there is no existing method that does both.

Answer Set Programming (ASP) is a non-monotonic logic programming language that is good at representing and reasoning with commonsense knowledge (Gelfond and Kahl 2014). ASP in its default form cannot reason with probabilities. A *non-monotonic probabilistic logic* (P-log) extends ASP by allowing both logical and probabilistic arguments in its reasoning (Baral, Gelfond, and Rushton 2009). However, ASP and its extensions are ill-equipped to plan toward maximizing long-term rewards under uncertainty. Partially observable Markov decision processes (POMDPs) generalize Markov decision processes (MDPs) by assuming the partial observability of underlying states (Kaelbling, Littman, and Cassandra 1998). POMDPs can model the nondeterministic state transitions and unreliable observations using probabilities, and plan toward maximizing long-term rewards under such uncertainties. However, POMDPs are not designed to reason about commonsense knowledge. Furthermore, from a practical perspective due to the computational complexity of solving POMDPs, it is necessary to limit the modeled state variables as much as possible.

This paper presents an algorithm called CORPP that stands for combining *COmmonsense Reasoning and Probabilistic Planning*. CORPP combines P-log with POMDPs to, for the first time, integrate reasoning with (logical and probabilistic) commonsense knowledge and planning under probabilistic uncertainty. The key idea is to calculate possible worlds and generate informative priors for POMDP-based planning by reasoning with logical and probabilistic commonsense knowledge. In so doing, the logical reasoning component is able to shield state variables from the POMDP that affect the priors, but that are irrelevant to the optimal policy given the prior. In solving a shopping request identification problem, experimental results show significant improvements on both efficiency and accuracy compared to existing approaches using only P-log or POMDPs.

2 The CORPP Algorithm

Both the possible worlds and POMDP states are described using the same set of domain attributes. We say an attribute e is partially observable, if e's value can only be (unreliably) observed using sensors. The values of attributes that are not partially observable can be specified by facts, defaults, or reasoning with values of other attributes. The value of an attribute can be *unknown*. For instance, attribute *current time* can be specified by facts. Similarly, identities of people as facts can be available but not always. *Current location* (of a robot) is partially observable, because self-localization relies on sensors; and the value of attribute within *if it is within working hours now* can be inferred from *current time*.

We propose algorithm CORPP for reasoning with commonsense and planning under uncertainty, as shown in Fig-

A longer version of this paper (with the same title) has been accepted by AAAI-15. Please cite that paper to reference this work.

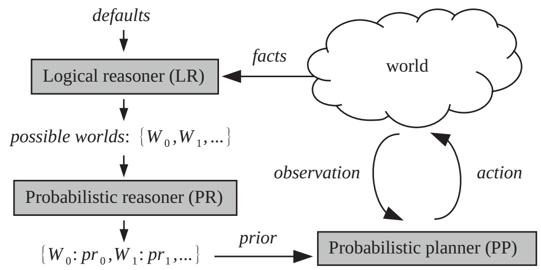

Figure 1: Overview of algorithm CORPP for combining commonsense reasoning with probabilistic planning

ure 1. The *logical reasoner* (LR) includes a set of logical rules in ASP and takes defaults and facts (Section 2.1) as input. The facts are collected by querying internal memory and databases. It is possible that facts and defaults try to assign values to the same attributes, in which case, default values will be automatically overwritten by facts. The output of LR is a set of possible worlds $\{W_0, W_1, \cdots\}$. Each possible world, as an answer set, includes a set of literals that specify the values of attributes—possibly unknown.

The *probabilistic reasoner* (PR) includes a set of random selection rules and probabilistic information assignments (Section 2.2) in P-log and takes the set of possible worlds as input. Reasoning with PR associates each possible world with a probability $\{W_0 : pr_0,\ W_1 : pr_1, \cdots\}$.

Unlike LR and PR, the *probabilistic planner* (PP), in the form of a POMDP, is specified by the goal of the task and the sensing and actuating capabilities of the agent (Section 2.3). The prior in Figure 1 is in the form of a distribution and denoted by α. The ith entry in the prior, α_i, is calculated by summing up the probabilities of possible worlds that are consistent with the corresponding POMDP state s_i. In practice, α_i is calculated by sending a P-log query of this form:

$$?\{\mathtt{s_i}\}|\mathtt{obs}(\mathtt{l_0}), \cdots, \mathtt{obs}(\mathtt{l_m}), \mathtt{do}(\mathtt{l_{m+1}}), \cdots, \mathtt{do}(\mathtt{l_n}).$$

where `l`'s are facts. If a fact `l` specifies the value of a random attribute, we use `obs(l)`. Otherwise we use `do(l)`. Technically, `do(l)` adds `l` into a program before calculating the possible worlds, while `obs(l)` is used to remove the calculated possible worlds that do not include literal `l`.

The prior is used for initializing POMDP beliefs in PP. Afterwards, the robot interacts with the world by continually selecting an action, executing the action, and making observations in the world. A task is finished after falling into a terminating state. CORPP is summarized in Algorithm 1. We next use an illustrative problem to present more details.

Illustrative Problem: *Shopping Request Identification* In a campus environment, the shopping robot can buy an item for a person and deliver to a room, so a shopping request is in the form of $\langle$*item, room, person*$\rangle$. A person can be either a *professor* or a *student*. Registered students are authorized to use the robot and professors are not unless they paid. The robot can get access to a database to query about registration and payment information, but the database may be incomplete. The robot can initiate spoken dialog to gather information for understanding shopping requests and take a delivery

Algorithm 1 The CORPP algorithm

Require: a task τ and a set of defaults $\mathcal{D}$
Require: a policy π produced by POMDP solvers
collect facts μ in the world, and add μ and $\mathcal{D}$ into LR
reason with LR and calculate possible worlds: $\mathcal{W}$
add the possible worlds into PR
for state $s_i \in \mathcal{S}$ **do**
  create a query ϕ_i using s_i and add ϕ_i into PR
  reason with PR, produce α_i, and remove ϕ_i from PR
end for
initialize belief state in PP: $b = \alpha$
repeat
  make an observation z; and update belief b
  select an action a using policy π
until s **is** *term*

action when it becomes confident in the estimation. This task is challenging for the robot because of its imperfect speech recognition ability. The goal is to identify shopping requests, e.g., $\langle$*coffee, office*1, *alice*$\rangle$, efficiently and robustly.

2.1 Logical Reasoning with ASP

Sorts and Objects: LR includes a set of sorts Θ : $\{\mathtt{time}, \mathtt{item}, \mathtt{room}, \mathtt{person}\}$ and a set of objects $\mathcal{O}$:

$$\mathtt{time} = \{\mathtt{morning}, \mathtt{noon}, \mathtt{afternoon}\}.$$
$$\mathtt{item} = \{\mathtt{sandwich}, \mathtt{coffee}\}.$$
$$\mathtt{room} = \{\mathtt{office1}, \mathtt{office2}, \mathtt{lab}, \mathtt{conference}\}.$$
$$\mathtt{person} = \{\mathtt{alice}, \mathtt{bob}, \mathtt{carol}, \mathtt{dan}, \mathtt{erin}\}.$$

Variables: We define the set of variables $\mathcal{V} : \{\mathtt{T}, \mathtt{I}, \mathtt{R}, \mathtt{P}\}$, using a construct `#domain`, which can be interpreted by popular ASP solvers.

```
#domain time(T).     #domain item(I).
#domain room(R).     #domain person(P).
```

Predicates: The set of predicates, $\mathcal{P}$, includes:

```
place(P,R). prof(P). student(P). registered(P).
authorized(P). paid(P). task(I,R,P).
```

where `place(P,R)` represents person `P`'s working room is `R`, `authorized(P)` states P is authorized to place orders, and a ground of `task(I,R,P)` specifies a shopping request.

The following two logical reasoning rules state that professors who have paid and students who have registered are authorized to place orders.

$$\mathtt{authorized(P)} \leftarrow \mathtt{paid(P)},\ \mathtt{prof(P)}.$$
$$\mathtt{authorized(P)} \leftarrow \mathtt{registered(P)},\ \mathtt{student(P)}.$$

Since the database can be incomplete about the registration and payment information, we need default knowledge to reason about unspecified variables. For instance, if it is unknown that a professor has paid, we believe the professor has not; if it is unknown that a student has registered, we believe the student has not.

$$\neg\mathtt{paid(P)} \leftarrow \mathtt{not\ paid(P)},\ \mathtt{prof(P)}.$$
$$\neg\mathtt{registered(P)} \leftarrow \mathtt{not\ registered(P)},\ \mathtt{student(P)}.$$

ASP is strong in default reasoning in that it allows prioritized defaults and exceptions at different levels (Gelfond and Kahl 2014). LR has the Closed World Assumption (CWA) for some predicates, e.g., the below rule guarantees that the value of attribute `authorized(P)` must be either `true` or `false` (cannot be unknown):

$$\neg\texttt{authorized(P)} \leftarrow \texttt{not authorized(P)}.$$

To identify a shopping request, the robot always starts with collecting all available facts, e.g.,

```
prof(alice). prof(bob). prof(carol). student(dan).
student(erin). place(alice, office1).
place(bob, office2). place(erin, lab).
```

If the robot also observes facts of `paid(alice)`, `paid(bob)` and `registered(dan)`, reasoning with the above defaults and rules will imply that `alice`, `bob` and `dan` are authorized to place orders. Thus, LR can generate a set of possible worlds by reasoning with the rules, facts and defaults.

2.2 Probabilistic Reasoning with P-log

PR includes a set of random selection rules describing possible values of random attributes:

$$\begin{array}{ll}\textbf{random}(\texttt{curr\_time}). & \texttt{curr\_time} : \texttt{time}.\\ \textbf{random}(\texttt{req\_item(P)}). & \texttt{req\_item} : \texttt{person} \rightarrow \texttt{item}.\\ \textbf{random}(\texttt{req\_room(P)}). & \texttt{req\_room} : \texttt{person} \rightarrow \texttt{room}.\\ \textbf{random}(\texttt{req\_person}). & \texttt{req\_person} : \texttt{person}.\end{array}$$

For instance, the second rule above states that if the delivery is for person `P`, the value of `req_item` is randomly selected from the range of `item`, unless fixed elsewhere. The following two *pr-atoms* state the probability of delivering for person `P` to `P`'s working place (0.8) and the probability of delivering `coffee` in the morning (0.8).

$$\texttt{pr(req\_room(P)} = \texttt{R} \mid \texttt{place(P,R))} = 0.8.$$
$$\texttt{pr(req\_item(P)} = \texttt{coffee} \mid \texttt{curr\_time} = \texttt{morning)} = 0.8.$$

Such random selection rules and pr-atoms allow us to represent and reason with commonsense with probabilities. Finally, a shopping request is specified as follows:

$$\begin{aligned}\texttt{task(I,R,P)} \leftarrow & \texttt{req\_item(P)} = \texttt{I},\ \texttt{req\_room(P)} = \texttt{R},\\ & \texttt{req\_person} = \texttt{P},\ \texttt{authorized(P)}.\end{aligned}$$

PR takes queries from PP and returns the joint probability. For instance, if it is known that Bob, as a professor, has paid and the current time is morning, a query for calculating the probability of $\langle\texttt{sandwich}, \texttt{office1}, \texttt{alice}\rangle$ is of the form:

$$\begin{aligned}?\{\texttt{task(sandwich,office1,alice)}\} \mid & \texttt{do(paid(bob))},\\ & \texttt{obs(curr\_time} = \texttt{morning)}.\end{aligned}$$

The fact of `bob` having paid increases the uncertainty in estimating the value of `req_person` by bringing additional possible worlds that include `req_person` = `bob`.

2.3 Probabilistic planning with POMDPs

A POMDP needs to model all partially observable attributes relevant to the task at hand. In the shopping request identification problem, an underlying state is composed of an item, a room and a person. The robot can ask polar questions such as "*Is this delivery for Alice?*", and wh-questions such as "*Who is this delivery for?*". The robot expects observations of "yes" or "no" after polar questions and an element from the sets of items, rooms, or persons after wh-questions. Once the robot becomes confident in the request estimation, it can take a delivery action that deterministically leads to a terminating state. Each delivery action specifies a shopping task.

- $\mathcal{S} : \mathcal{S}_i \times \mathcal{S}_r \times \mathcal{S}_p \cup$ *term* is the state set. It includes a Cartesian product of the set of items $\mathcal{S}_i$, the set of rooms $\mathcal{S}_r$, and the set of persons $\mathcal{S}_p$, and a terminal state *term*.
- $\mathcal{A} : \mathcal{A}_w \cup \mathcal{A}_p \cup \mathcal{A}_d$ is the action set. $\mathcal{A}_w = \{a^i_w, a^r_w, a^p_w\}$ includes actions of asking wh-questions. $\mathcal{A}_p = \mathcal{A}^i_p \cup \mathcal{A}^r_p \cup \mathcal{A}^p_p$ includes actions of asking polar questions, where $\mathcal{A}^i_p$, $\mathcal{A}^r_p$ and $\mathcal{A}^p_p$ are the sets of actions of asking about items, rooms and persons respectively. $\mathcal{A}_d$ includes the set of delivery actions. For $a \in \mathcal{A}_d$, we use $s \odot a$ to represent that the delivery of a matches the underlying state s (i.e., a correct delivery) and use $s \oslash a$ otherwise.
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the state transition function. Action $a \in \mathcal{A}_w \cup \mathcal{A}_p$ does not change the state and action $a \in \mathcal{A}_d$ results in the terminal state *term* deterministically.
- $Z : Z_i \cup Z_r \cup Z_p \cup \{z^+, z^-\}$ is the set of observations, where Z_i, Z_r and Z_p include observations of action *item*, *room* and *person* respectively. z^+ and z^- are the positive and negative observations after polar questions.
- $O : \mathcal{S} \times \mathcal{A} \times Z \rightarrow [0,1]$ is the observation function. The probabilities of O are empirically hand-coded, e.g., z^+ and z^- are more reliable than other observations. Learning the probabilities is beyond the scope of this paper.
- $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function. In our case:

  $$R(s,a) = \begin{cases} -r_p, & \textbf{if } s \in \mathcal{S},\ a \in \mathcal{A}_p \\ -r_w, & \textbf{if } s \in \mathcal{S},\ a \in \mathcal{A}_w \\ -r^-_d, & \textbf{if } s \in \mathcal{S},\ a \in \mathcal{A}_d,\ s \oslash a \\ r^+_d, & \textbf{if } s \in \mathcal{S},\ a \in \mathcal{A}_d,\ s \odot a \end{cases} \quad (1)$$

  where we use r_w and r_p to specify the costs of asking wh- and polar questions. r^-_d is a big cost for an incorrect delivery and r^+_d is a big reward for a correct one. Unless otherwise specified, $r_w = 1$, $r_p = 2$, $r^-_d = 100$, and $r^+_d = 50$.

Given a POMDP, we calculate a policy using state-of-the-art POMDP solvers, e.g., APPL (Kurniawati, Hsu, and Lee 2008). The policy maps a POMDP belief to an action toward maximizing the long-term rewards. Specifically, the policy enables the robot to take a delivery action only if it is confident enough about the shopping request that the cost of asking additional questions is not worth the expected increase in confidence. The policy also decides *what*, for *whom* and *where* to deliver. There are attributes that contribute to calculating the POMDP priors but are irrelevant to the optimal policy given the prior. The reasoning components shield such attributes, e.g., `curr_time`, from the POMDPs.

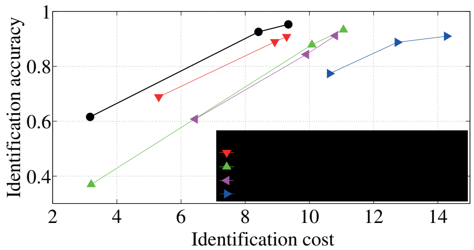

Figure 2: CORPP performs better than the other approaches in both efficiency and accuracy (Hypothesis-III). Each data point is the average of at least 10,000 simulated trials.

3 Experiments

The experiments focus on comparisons based on results of large numbers of simulated trials, and were designed to evaluate the hypothesis that CORPP performs the best in both accuracy and efficiency by combining LR, PR, and PP.

Probabilistic Knowledge at Different Levels: The robot does not necessarily have full and/or accurate probabilistic commonsense knowledge. We distinguish the probabilistic knowledge provided to the robot based on its availability and accuracy. **All**: the robot can get access to the knowledge described in Section 2.1 and 2.2 in a complete and accurate way. **Limited**: the accessibility to the knowledge is the same as "All" except that current time is hidden from the robot. **Inaccurate**: the accessibility to the knowledge is the same as "All" except that the value of current time is always wrong.

We provide the probabilistic commonsense knowledge (Section 2.2) to the robot at different completeness and accuracy levels—learning the probabilities is beyond the scope of this paper. Experimental results are shown in Figure 2. Each set of experiments has three data points because we assigned different penalties to incorrect identifications in PP (r_d^- equals 10, 60 and 100). Generally, a larger penalty requires the robot to ask more questions before taking a delivery action. POMDP-based PP without commonsense reasoning (blue rightward triangle) produced the worst results. Combining LR with PP (magenta leftward triangle) improves the performance by reducing the number of possible worlds. Giving *inaccurate* probabilistic commonsense (green upward triangle) significantly hurts the accuracy of CORPP when the penalty of incorrect identifications is small. CORPP with *limited* probabilistic commonsense requires much less cost and results in higher (or at least similar) accuracy on average, compared to planning without PR. Finally, CORPP with *all* knowledge produced the best performance in both efficiency and accuracy.

4 Related Work

Researchers have developed algorithms and frameworks that combine logical and probabilistic reasoning, e.g., probabilistic first-order logic (Halpern 2003) and Markov logic network (Richardson and Domingos 2006). However, algorithms based on first-order logic for probabilistic reasoning have difficulties in representing or reasoning with commonsense. P-log (Baral, Gelfond, and Rushton 2009) can do logical and probabilistic reasoning with commonsense but has difficulties to plan toward maximizing long-term rewards.

POMDPs have been applied to a variety of probabilistic planning tasks (Young et al. 2013; Zhang, Sridharan, and Washington 2013). However, existing POMDP-based planning work does not readily support representation of or reasoning with rich commonsense knowledge. Furthermore, from a practical perspective, the state variables modeled by POMDPs have to be limited to allow real-time operation. This makes it challenging to use POMDPs in large, complex state-action spaces, even if hierarchical decomposition and approximate algorithms have been applied (Zhang, Sridharan, and Washington 2013; Kurniawati, Hsu, and Lee 2008).

Existing work investigated generating priors by ASP-based inference for POMDP-based planning (Zhang, Sridharan, and Bao 2012). However, that work did not have a probabilistic reasoner to reason with probabilistic commonsense knowledge. Furthermore, the logical reasoner in that work did not calculate possible worlds for POMDPs.

5 Conclusions

This paper presents the CORPP algorithm that integrates reasoning with commonsense knowledge and planning under probabilistic uncertainty. Answer Set Programming, a non-monotonic logic programming language, is used to reason with *logical* commonsense knowledge. P-log, a probabilistic extension of ASP, further enables reasoning with *probabilistic* commonsense knowledge. POMDPs are used to plan under uncertainty toward maximizing long-term rewards. The complementary features of ASP and POMDPs ensure efficient, robust information gathering and behavior in robotics. Experimental results on a shopping request identification problem show significant improvements on both efficiency and accuracy, compared with existing approaches using P-log or POMDPs individually.

6 Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-1330072, CNS-1305287), ONR (21C184-01), AFOSR (FA8750-14-1-0070, FA9550-14-1-0087), and Yujin Robot.

References

Baral, C.; Gelfond, M.; and Rushton, N. 2009. Probabilistic Reasoning with Answer Sets. *Theory and Practice of Logic Programming* 9(1):57–144.

Gelfond, M., and Kahl, Y. 2014. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach.* Cambridge University Press.

Halpern, J. Y. 2003. Reasoning about Uncertainty.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* 101:99–134.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. In *Robotics: Science and Systems*.

Richardson, M., and Domingos, P. 2006. Markov Logic Networks. *Machine Learning* 62(1-2):107–136.

Young, S.; Gasic, M.; Thomson, B.; and Williams, J. D. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE* 101(5):1160–1179.

Zhang, S.; Sridharan, M.; and Bao, F. S. 2012. ASP+POMDP: Integrating Non-monotonic Logic Programming and Probabilistic Planning on Robots. In *International Conference on Development and Learning (ICDL)*.

Zhang, S.; Sridharan, M.; and Washington, C. 2013. Active Visual Planning for Mobile Robot Teams Using Hierarchical POMDPs. In *IEEE Transactions on Robotics*.