Active Generation of Logical Rules for POMCP Shielding

Extended Abstract

Giulio Mazzi
Università degli Studi di Verona
Verona, Italy
giulio.mazzi@univr.it

Alberto Castellini
Università degli Studi di Verona
Verona, Italy
alberto.castellini@univr.it

Alessandro Farinelli
Università degli Studi di Verona
Verona, Italy
alessandro.farinelli@univr.it

ABSTRACT

We consider the popular Partially Observable Monte-Carlo Planning (POMCP) algorithm and propose a methodology, called *Active XPOMCP*, for generating compact logical rules that represent properties of the control policy. These rules are then used as shields to prevent POMCP from selecting unexpected actions, with useful implications on the security and trustworthiness of the algorithm. Contrary to state-of-the-art methods, *Active XPOMCP* does not require a previously generated set of belief-action pairs to generate the logical rule, but it actively generates this data in an information-efficient way by querying the algorithm. *Active XPOMCP* reduces the number of beliefs needed to generate accurate rules with respect to state-of-the-art methods, and it allows to produce more accurate shields when few belief-action samples are available.

KEYWORDS

POMDP; POMCP; Shielding; SMT

ACM Reference Format:
Giulio Mazzi, Alberto Castellini, and Alessandro Farinelli. 2022. Active Generation of Logical Rules for POMCP Shielding: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Safety and trustworthiness are very important topics in AI [5]. In this work, we focus on the safety of a popular model-based RL algorithm called Partially Observable Monte Carlo Planning (POMCP) [9]. POMCP can scale to large instances thanks to its online and approximate nature. However, the generated policies are very difficult to analyze [4]. Our approach aims to improve the safety of POMCP via shielding [1, 2]. Specifically, we generate logical rules (e.g., "the robot should move fast if its confidence to be in a lowly cluttered path is higher than 90%") starting from rule templates (e.g., "the robot should move fast if its confidence to be in a lowly cluttered path is higher than x%", where x is a free variable) written by humans, which specify properties of interest of the policy. We assume that representing properties related to safety is simpler than representing the entire policy, hence safety properties can be represented by human-understandable logical rules. Once a rule template has been written, the free variables in it are instantiated by observing the behaviour of POMCP. This instantiation is computed using the beliefs of an agent (i.e., points in the space of probability distributions over states). This space (a simplex) is continuous and huge in real-world problems. However, given a specific model of the environment and an initial belief, only a subset of the beliefs is reachable [8]. The main contribution of this work is an active strategy, called *Active XPOMCP*, for sampling the belief space of POMCP efficiently to generate accurate logical rules using a small number of informative and reachable belief samples. In the example above, these belief samples should be located as close as possible to the 90% confidence of being in a lowly cluttered path (i.e., the true decision boundary of the policy). Our strategy explores beliefs efficiently, moving in reachable points of the belief space. *Active XPOMCP* formalizes this search as a maximum satisfiability modulo theory (MAX-SMT) problem. This allows to express complex logical formulas and compute optimal assignments when the template is not fully satisfiable (which happens in most real-world cases). The active search strategy is based on the definition of an *uncertainty interval* which identifies, by two logical rules, parts of the belief space currently unexplored. New belief samples are selected inside the uncertainty interval to reduce the uncertainty of the rules. The rules are then used as a shield to prevent POMCP from selecting unwanted actions. Results show that *Active XPOMCP* outperforms the state-of-the-art non-active strategy XPOMCP [6, 7] in a domain (called *velocity regulation* in the following) in which the velocity of a mobile robot has to be tuned to avoid collisions with obstacles [4]. Standard XPOMCP uses a predefined trace generated by executing POMCP without any strategy to search informative beliefs. We show that *Active XPOMCP* manages to reduce the uncertainty interval quicker than XPOMCP and, consequently, it generates accurate rules using fewer *runs* (a *run* is a complete execution of POMCP). Furthermore, we show that the rules generated by *Active XPOMCP* are more accurate than those produced by XPOMCP when the belief space is large.

2 OVERVIEW OF ACTIVE XPOMCP

Active XPOMCP builds logical formulas that describe properties of POMCP policies in a human-comprehensible way. Human experts design the structure of these formulas to embed assumptions about the policy. The rules produced by our method are then used as shields in POMCP, improving its safety. The main difference between the state-of-the-art approach, namely XPOMCP [6, 7], and *Active XPOMCP* is that our approach does not describe a set of executions; instead, it uses the policy directly to find relevant information. *Active XPOMCP* queries a POMCP instance, providing observations and receiving actions, with the aim to minimize the number of queries required to generate accurate rules. In detail, since a rule can only describe the policy according to the beliefs sampled by *Active XPOMCP*, we represent the *uncertainty* of a rule as a set of unexplored beliefs. The rule uncertainty is a key concept used to guide two processes, namely, the identification of the most

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online. © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

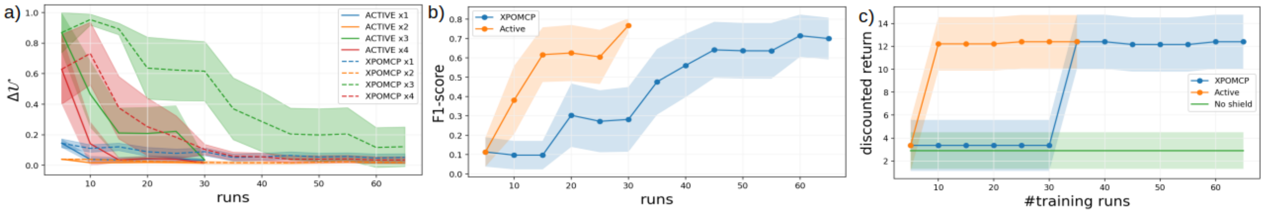

Figure 1: Results on *velocity regulation a.* mean uncertainty interval $\Delta\mathcal{U}$; *b.* mean F_1 score; *c.* average discounted return.

informative beliefs and the termination of the rule synthesis procedure. The rule uncertainty depends on the fact that a template could have multiple solutions because the trace contains only partial information about the desired property. A rule is located in a position that satisfies the majority of the beliefs in the current trace; hence several positions are available if the beliefs are far from the decision boundary. XPOMCP locates the rule as close as possible to the beliefs (in the trace) related to the rule action while *Active XPOMCP* reduces the uncertainty interval in order to locate the rule as close as possible to the decision boundary. *Active XPOMCP* computes two rules to describe the behaviour of the policy on a specific action. The first rule is called *strict rule* and is located as close as possible to the observed beliefs related to that action. The second rule is called *loose rule*, and it is located as far as possible from the beliefs related to the action, without including beliefs related to other actions. The strict rule behaves conservatively and describes only beliefs collected in the trace about the action of interest, and the loose rule captures all the beliefs (explored or not) that do not explicitly violate the requirement expressed by the expert in the rule template. *Active XPOMCP* reduces the distance between the two rules by selecting informative beliefs in the uncertainty interval, and this decreases the distance between loose and the strict rule until the two converge. This is non-trivial in real problems because the belief space is highly multidimensional. Since only a small part of the belief space is reachable from the initial belief, considering only *reachable* beliefs reduces the search space and help scalability.

3 RESULTS

We test *Active XPOMCP* in the *velocity regulation* domain, and compare it with XPOMCP. Results show that it generates more accurate rules using fewer runs than XPOMCP.

In *velocity regulation* [3], a robot travels on a path divided into eight *segments* and *subsegments.* Each segment has a (hidden) difficulty value among *clear*, *lightly cluttered* or *heavily cluttered* (i.e., the state-space has 3^8 states). The goal of the robot is to travel on this path as fast as possible while avoiding collisions. In each subsegment, the robot must decide a *speed level* $a \in \{0, 1, 2\}$. Higher speed levels yield higher rewards and greater risks of collision. After each subsegment, the robot receives a noisy observation on the current difficulty. We consider a rule that describes when the robot should move at high speed. We expect the robot to move fast only if it is confident that the current segment is easy to navigate:

$$\texttt{select 2} \iff p(0) \geq x_1 \lor p(1) \leq x_2 \lor (p(0) \geq x_3 \land p(1) \geq x_4).$$

It specifies that the agent should move fast if at least one of three conditions is satisfied, namely, *i)* its confidence of being in a clear segment is above a threshold ($p(0) \geq x_1$), *ii)* its confidence of being in a heavily cluttered segment is below a threshold ($p(2) \leq x_2$), *iii)* the combination of $p(0)$ and $p(1)$ is above two thresholds ($p(0) \geq x_3 \land p(1) \geq x_4$).

First, we analyze how the uncertainty interval $\Delta\mathcal{U}$ depends on the number of runs. Figure 1.a presents the size of uncertainty interval $\Delta\mathcal{U}$ of each variable x_1, x_2, x_3, x_4, separately, using $N \in \{5, 10, \ldots, 65\}$ runs. The test is repeated ten times with different seeds. The interval sizes decrease both for rules computed by XPOMCP and for rules computed by *Active XPOMCP*, but the active approach is significantly faster. Figure 1.b presents how accurately these rules describe the policy (measured as F_1 score). The F_1 score is heavily impacted by the change in $\Delta\mathcal{U}$. XPOMCP requires 60 or more runs, on average, to achieve performance comparable to that reached by *Active XPOMCP* with 10 runs. In particular, the difference is large for variable x_3 and x_4. This is important because a poor instantiation of these variables leads to poor shielding performance. The rules generated by *Active XPOMCP* reach a maximum F_1 score of 0.79, which is higher than the F_1 score reached by rules generated by XPOMCP (i.e., 0.71) but lower than the maximum value 1.0. This is because the rules generate some false negatives (i.e., the rules predict that the robot moves fast, but it moves slowly).

Then, we measure the shielding performance using the same N (Figure 1.c). For each value of N we compute a logical rule using XPOMCP and another rule using *Active XPOMCP*. In the second case, the algorithm starts from $N = 5$, and it converges when the uncertainty interval of each variable is lower than a threshold $\epsilon = 0.02$. We evaluate the shielding performance of the rules generated using the two methodologies, and the mechanism presented in [7]. Figure 1.c shows the average discounted return of the shielded POMCP. *Active XPOMCP* achieves the maximum performance using only ten runs, while XPOMCP reaches this value only after 35 runs. Between 10 and 30 runs, *Active XPOMCP* achieves a relative performance increase of up to 264% compared to XPOMCP. This is because the last part of the rule (i.e., $p(0) \geq x_3 \land p(1) \geq x_4$) is hard to tune but important to describe the behaviour of the agent. We run the experiments using an Intel Core i7-6700HQ and 16GB RAM. The code is available at *https://github.com/GiuMaz/XPOMCP*.

4 CONCLUSIONS

This article presents a methodology that builds a logical description of a POMCP policy by actively exploring the belief space. Our experiments showed that the active approach builds precise rules using significantly less information than state-of-the-art methods.

REFERENCES

[1] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. 2018. Safe Reinforcement Learning via Shielding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 2669–2678.

[2] Roderick Bloem, Bettina Könighofer, Robert Könighofer, and Chao Wang. 2015. Shield Synthesis: Runtime Enforcement for Reactive Systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, Christel Baier and Cesare Tinelli (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 533–548.

[3] Alberto Castellini, Georgios Chalkiadakis, and Alessandro Farinelli. 2019. Influence of State-Variable Constraints on Partially Observable Monte Carlo Planning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 5540–5546.

[4] Alberto Castellini, Enrico Marchesini, Giulio Mazzi, and Alessandro Farinelli. 2020. Explaining the Influence of Prior Knowledge on POMCP Policies. In *Multi-Agent Systems and Agreement Technologies - 17th European Conference, EUMAS 2020, and 7th International Conference, AT 2020, Thessaloniki, Greece, September 14-15, 2020, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 12520)*, Nick Bassiliades, Georgios Chalkiadakis, and Dave de Jonge (Eds.). Springer, 261–276.

[5] High-Level Expert Group on AI. 2019. *Ethics guidelines for trustworthy AI.* Report. European Commission. https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

[6] Giulio Mazzi, Alberto Castellini, and Alessandro Farinelli. 2021. Identification of Unexpected Decisions in Partially Observable Monte-Carlo Planning: A Rule-Based Approach. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé (Eds.). ACM, 889–897.

[7] Giulio Mazzi, Alberto Castellini, and Alessandro Farinelli. 2021. Rule-based Shielding for Partially Observable Monte-Carlo Planning. In *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling, ICAPS 2021, Guangzhou, China (virtual), August 2-13, 2021*, Susanne Biundo, Minh Do, Robert Goldman, Michael Katz, Qiang Yang, and Hankz Hankui Zhuo (Eds.). AAAI Press, 243–251.

[8] Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, Georg Gottlob and Toby Walsh (Eds.). Morgan Kaufmann, 1025–1032.

[9] David Silver and Joel Veness. 2010. Monte-Carlo Planning in Large POMDPs. In *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe Taylor, R. S. Zemel, and A. Culotta (Eds.). Curran Associates, Inc., 2164–2172. http://papers.nips.cc/paper/4031-monte-carlo-planning-in-large-pomdps.pdf