Methods and Mechanisms for Interactive Novelty Handling in Adversarial Environments

Extended Abstract

Tung Thai
Tufts University
Medford, MA, United States
tung.thai@tufts.edu

Mudit Verma
Arizona State University
Tempe, AZ, United States
mverma13@asu.edu

Utkarsh Soni
Arizona State University
Tempe, AZ, United States
usoni1@asu.edu

Sriram Gopalakrishnan
Arizona State University
Tempe, AZ, United States
sgopal28@asu.edu

Ming Shen
Arizona State University
Tempe, AZ, United States
mshen16@asu.edu

Mayank Garg
Arizona State University
Tempe, AZ, United States
mgarg20@asu.edu

Ayush Kalani
Arizona State University
Tempe, AZ, United States
akalani2@asu.edu

Nakul Vaidya
Arizona State University
Tempe, AZ, United States
nvaidya7@asu.edu

Neeraj Varshney
Arizona State University
Tempe, AZ, United States
nvarshn2@asu.edu

Chitta Baral
Arizona State University
Tempe, AZ, United States
chitta@asu.edu

Subbarao Kambhampati
Arizona State University
Tempe, AZ, United States
rao@asu.edu

Jivko Sinapov
Tufts University
Medford, MA, United States
jivko.sinapov@tufts.edu

Matthias Scheutz
Tufts University
Medford, MA, United States
matthias.scheutz@tufts.edu

ABSTRACT

Learning to detect, characterize and accommodate novelties is a challenge that agents operating in open-world domains need to address to achieve satisfactory task performance. We sketch general methods for detecting and characterizing different types of novelties, and for building an appropriate adaptive model to accommodate them utilizing logical representations and reasoning methods in stochastic partially observable multi-agent environments. We also briefly report results from evaluations of our algorithms in the game domain of Monopoly. The results show high novelty detection and accommodation rates.

KEYWORDS

Open-world AI, Agent Architecture, Adaptive Multiagent Systems

ACM Reference Format:
Tung Thai, Mudit Verma, Utkarsh Soni, Sriram Gopalakrishnan, Ming Shen, Mayank Garg, Ayush Kalani, Nakul Vaidya, Neeraj Varshney, Chitta Baral, Subbarao Kambhampati, Jivko Sinapov, and Matthias Scheutz. 2023. Methods and Mechanisms for Interactive Novelty Handling in Adversarial Environments : Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

1 INTRODUCTION: OPEN-WORLD AI

Many classical AI tasks take place in *closed-world* domains where the types of entities, their actions, and the overall domain dynamics are known. In contrast, *open-world domains* allow for novel entities, actions, etc. to arise anytime unbeknownst to the task-performing agent who needs to handle them (cp. to [1, 18]). Especially *interactive novelties* where agents interact with each other and with the environment in novel ways present a challenge for agents departing from a *closed-world* assumption (e.g., [2, 6, 9]. This is different from the agent being unaware about certain parts of the world , like other agent's rewards [11, 14–17] or reasoning mechanisms [3–5, 12], as opposed to the world-changing without emitting explicit signals to the agent.

To tackle the challenges of interactive novelties in open-world environments, we developed a general novelty-handling framework that uses symbolic logical reasoning to detect, learn, and adapt to novelties in *open-world* environments. The results suggest that our agent can detect novelty with a high accuracy rate while maintaining a dominant performance against other game-playing agents. For a more complete description of our work, please see our full paper on arXiv [13].

2 METHODS AND MECHANISMS

We will use the multi-player adversarial board game Monopoly to briefly describe our methods for detecting, characterizing, and accommodating novelties, as it was also used for the evaluation. In Monopoly, up to four players roll dice to make moves and take actions on the game board with the goal of being the last player standing after bankrupting other players. This objective can be achieved by buying properties, monopolizing color sets, and developing houses on properties. The game includes different surprise factors such as chance cards, community cards, jail, auction, and trading ability between agents. Hence, any action in the game needs to be adapted to dice rolls, community cards, chance cards, and the decisions of other players. Unlike traditional Monopoly, where one can fully observe all the states and actions of other agents, the "Open-world Monopoly" version is only partially observable, i.e., it does not allow us to monitor all the actions and interactions on our turn [7].

2.1 Novelty Detection

We record the information of the game as provided by the Monopoly simulation ("game environment") and compare it with our "expectation" state of the game board. This "expectation" state is derived from the agent's knowledge base of the game, including expected states, actions, action preconditions, and end effects. Then, the game environment provides us with the actual game board states and actions that have occurred between the current time step and the previous time our agent performed an action. When we notice a discrepancy between our expected state and the actual state, we surmise that something must have changed within the game.

2.2 Novelty Characterization

Next, the agent uses a novelty identification module to characterize the novelty using "Answer Set Programming" (ASP). The resulting program's answer sets give us the parameter values which reconcile the predicted game board state and the observed game board state. If there is only one answer set and thus a unique parameter value, then if this value is different from the value we had earlier, we have identified a novelty. Now we can update our ASP code that was used for hypothetical reasoning by simply replacing the earlier value of the parameter with the new value.

2.3 Novelty Accommodation

Since novelties in the state (features, dynamics, actions) mean the agent would have to replan often and would have to do so based on the most updated information, we were interested in developing an online planning algorithm to determine the best action. However, with environments that are both *long-horizon* and *stochastic*, using online planning approaches like Monte-Carlo tree search, quickly becomes intractable. To address this problem, we formulate a truncated-rollout-based algorithm that uses updated domain dynamics (learned from detected novelties) for a few steps of the rollout and then uses a state evaluation function to approximate the return for the rest of that rollout. In our evaluation function, we use both domain-specific components and a more general heuristic to approximate the return from the state after the truncated rollout.

Action Novelties			
TPR	100%	100%	100%
FPR	0%	0%	0%
NRP	151.79%	135.38%	143.08%
Interaction Novelties			
TPR	100%	100%	100%
FPR	0%	0%	0%
NRP	130.46%	134.15%	113.23%
Relation Novelties			
TPR	100%	100%	80%
FPR	0%	0%	0%
NRP	146.46%	121.85%	145.23%

Table 1: Evaluation results (see text for details).

Furthermore, to ensure the agent adapts to the detected novelties, we made both the environment simulator used for rollouts and the evaluation function sufficiently flexible and conditioned on the environment attributes; we only used a few tuned constants. Thus, whenever a novelty was detected, we updated the relevant attributes in our simulator and evaluation function before running our algorithm to decide our actions. Using this approach, we are able to incorporate novel information into our decision-making process and adapt efficiently.

3 EVALUATION & RESULTS

In an effort to maintain the integrity of the evaluation, all the information about the novelty was hidden from our team, and all the information about our architecture or methodologies was also hidden from the evaluation team. Our agent was evaluated based on three different metrics: the correctly detect novelties, i.e., true positive rate (TPR), the incorrectly detect novelties, i.e., false positive rate (FPR), and the novelty reaction performance (NRP) after the novelty was introduced (post-novelty) in Table 1. We compute the novelty reaction performance (NRP) of the agent based on the following formula: $NRP = \frac{\mathcal{W}_{agent}}{\mathcal{W}_{baseline}}$ where, $\mathcal{W}_{agent}$ is the win rate of our agent. $\mathcal{W}_{baseline}$ is 65%.

4 CONCLUSION

Our work presented a new agent architecture for interactive novelty handling in an adversarial environment that can detect, characterize, and accommodate novelties. First, we use ASP to detect and characterize interactive novelties. Then, we update the detected novelties to our agent's knowledge base. Finally, we utilize the truncated-rollout MCTS agent to accommodate the novelty. The external evaluation results support the cognitive architecture's effectiveness in handling different levels of interactive novelty. In the near future, we would like to model the opponents' behavior using reinforcement learning due to its potential to learn opponents' behavior without knowing opponent's observations and actions [8, 10]. Ultimately, we believe improving the model's capability of predicting another agent's behaviors is the biggest area for growth.

ACKNOWLEDGMENTS

This work was funded in part by DARPA grant W911NF-20-2-0006. We would like to thank Mayank Kejriwal, Shilpa Thomas, Min-Hsueh Chiu and other members of the University of Southern California team for the Monopoly simulator and agent evaluation.

REFERENCES

[1] Rinu Boney, Alexander Ilin, Juho Kannala, and Jarno Seppanen. 2021. Learning to Play Imperfect-Information Games by Imitating an Oracle Planner. *IEEE Transactions on Games* (2021). https://doi.org/10.1109/TG.2021.3067723

[2] Noam Brown and Tuomas Sandholm. 2019. Solving Imperfect-Information Games via Discounted Regret Minimization. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 1829–1836. https://doi.org/10.1609/aaai.v33i01.33011829

[3] Sriram Gopalakrishnan, Mudit Verma, and Subbarao Kambhampati. 2021. Computing Policies That Account For The Effects Of Human Agent Uncertainty During Execution In Markov Decision Processes. *arXiv preprint arXiv:2109.07436* (2021).

[4] Sriram Gopalakrishnan, Mudit Verma, and Subbarao Kambhampati. 2021. Synthesizing Policies That Account For Human Execution Errors Caused By State Aliasing In Markov Decision Processes. In *ICAPS 2021 Workshop on Explainable AI Planning URL https://openreview. net/pdf.*

[5] Lin Guan*, Mudit Verma*, and Subbarao Kambhampati. 2020. Explanation augmented feedback in human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2006.14804* (2020).

[6] Johannes Heinrich and David Silver. 2016. Deep Reinforcement Learning from Self-Play in Imperfect-Information Games. *ArXiv: 1603.01121* (2016). arXiv:1603.01121 http://arxiv.org/abs/1603.01121

[7] Mayank Kejriwal and Shilpa Thomas. 2021. A multi-agent simulator for generating novelty in monopoly. *Simulation Modelling Practice and Theory* 112 (2021), 102364. https://doi.org/10.1016/j.simpat.2021.102364

[8] Georgios Papoudakis and Stefano V. Albrecht. 2020. Variational Autoencoders for Opponent Modeling in Multi-Agent Systems. *CoRR* abs/2001.10829 (2020). arXiv:2001.10829 https://arxiv.org/abs/2001.10829

[9] Marc Ponsen, Pieter Spronck, Héctor Muñoz-Avila, and David W. Aha. 2007. Knowledge acquisition for adaptive game AI. *Science of Computer Programming* 67, 1 (2007), 59–75. https://doi.org/10.1016/j.scico.2007.01.006 Special Issue on Aspects of Game Programming.

[10] Roxana Radulescu, Timothy Verstraeten, Yijie Zhang, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. 2020. Opponent Learning Awareness and Modelling in Multi-Objective Normal Form Games. *CoRR* abs/2011.07290 (2020). arXiv:2011.07290 https://arxiv.org/abs/2011.07290

[11] Utkarsh Soni, Sarath Sreedharan, Mudit Verma, Lin Guan, Matthew Marquez, and Subbarao Kambhampati. 2022. Towards customizable reinforcement learning agents: Enabling preference specification through online vocabulary expansion. https://doi.org/10.48550/ARXIV.2210.15096

[12] Sarath Sreedharan, Utkarsh Soni, Mudit Verma, Siddharth Srivastava, and Subbarao Kambhampati. 2020. Bridging the Gap: Providing Post-Hoc Symbolic Explanations for Sequential Decision-Making Problems with Inscrutable Representations. *arXiv preprint arXiv:2002.01080* (2020).

[13] Tung Thai, Ming Shen, Mayank Garg, Ayush Kalani, Nakul Vaidya, Utkarsh Soni, Mudit Verma, Sriram Gopalakrishnan, Chitta Baral, Subbarao Kambhampati, Jivko Sinapov, and Matthias Scheutz. 2023. Methods and Mechanisms for Interactive Novelty Handling in Adversarial Environments. https://doi.org/10.48550/ARXIV.2302.14208

[14] Mudit Verma, Siddhant Bhambri, and Subbarao Kambhampati. 2023. Exploiting Unlabeled Data for Feedback Efficient Human Preference based Reinforcement Learning. *arXiv preprint arXiv:2302.08738* (2023).

[15] Mudit Verma and Subbarao Kambhampati. 2023. Data Driven Reward Initialization for Preference based Reinforcement Learning. *arXiv preprint arXiv:2302.08733* (2023).

[16] Mudit Verma and Subbarao Kambhampati. 2023. A State Augmentation based approach to Reinforcement Learning from Human Preferences. *arXiv preprint arXiv:2302.08734* (2023).

[17] Mudit Verma and Katherine Metcalf. 2022. Symbol Guided Hindsight Priors for Reward Learning from Human Preferences. *arXiv preprint arXiv:2210.09151* (2022).

[18] Tezira Wanyana and Deshendran Moodley. 2021. *An Agent Architecture for Knowledge Discovery and Evolution.* 241–256. https://doi.org/10.1007/978-3-030-87626-5_18