Efficient Collaboration with Unknown Agents: Ignoring Similar Agents without Checking Similarity

Extended Abstract

Yansong Li
University of Illinois Chicago
Chicago, United States
yli340@uic.edu

Shuo Han
University of Illinois Chicago
Chicago, United States
hanshuo@uic.edu

ABSTRACT

Ad hoc teamwork (AHT) is concerned with developing an AI agent who learns to collaborate with different previously unseen partners. We consider a setting where the AI agent is provided with a hypothesis set of partners' policies. Several online algorithms that take the hypothesis set as input can be applied to solve the AHT problem. One way to speed up these online learning algorithms is to eliminate the redundant policies, i.e., partner models sharing the same collaborating policy, from the hypothesis set. Nevertheless, we show whether this elimination should be applied depends on the learning algorithm used by the AI agent. Specifically, we identify a property of a learning algorithm: *redundancy-aware*. When the learning algorithm is redundancy-aware, redundancy elimination is unnecessary. In other words, redundancy-aware algorithms can ignore similar agents in the hypothesis set. We demonstrate through an example that an online algorithm with redundancy-aware property exists when the hypothesis set contains the true partner policy. We test our approach on a team Markov game of two players. Comparative numerical analyses reveal that the redundancy-aware algorithm outperforms other standard no-regret learning algorithms including upper confidence bound (UCB), Q-learning with UCB exploration, and the optimistic posterior sampling algorithm when the set of partner policies contains many redundant policies.

CCS CONCEPTS

• **Computing methodologies → Online learning settings**; *Multiagent reinforcement learning*; **Sequential decision making**.

KEYWORDS

Ad hoc teamwork; Online learning; Game theory

ACM Reference Format:
Yansong Li and Shuo Han. 2024. Efficient Collaboration with Unknown Agents: Ignoring Similar Agents without Checking Similarity: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Interaction with unknown agents is a vital capability of an AI agent. For example, autonomous vehicles must interpret and react to the movements of other vehicles on the road, all of which are unknown entities with potentially varying driving patterns. Without knowing the exact policy adopted by the partner, the AI agent cannot determine the environment in which it operates. Furthermore, the AI agent cannot assume that all partners take an optimal policy, especially when collaborating with human agents.

Ad hoc teamwork (AHT) [1, 6, 7] is the problem of developing an AI agent capable of collaborating with previously unseen partners, i.e., agents without prior coordination with the AI agent such as shared tasks, communication protocols, and joint training.

The AHT problem considered in this paper is a team Markov game involving two players: an AI agent under our control and a partner with an uncontrolled policy. The potential partner's policy belongs to an unknown set $\mathcal{H}^*$. The AI agent starts with a hypothesis set $\mathcal{H}$ that approximates $\mathcal{H}^*$. An online learning algorithm takes $\mathcal{H}$ as the input and generates a series of policies for the AI agent during episodic interactions with the unknown partner. The online algorithm solves the AHT problem if its regret is sublinear.

A method to speed up the online learning is to reduce the size of $\mathcal{H}$ while maintaining as many distinct partners, characterized by partner *types*. This method is called redundancy elimination. In this work, we show the existence of an online algorithm whose sample complexity only depends on the number of partner types instead of the number of partner policies. Thus, with this algorithm, redundancy elimination is unnecessary. To be more clear, we call online algorithms with such a property *redundancy-aware* algorithms. To show the existence of redundancy-aware algorithms, we take a specific algorithm called the Maximize to Explore (MEX) developed in [5] as an example. We show that the MEX algorithm is redundancy-aware both in theory and in practice. Therefore, there is no need to eliminate every redundant policy in $\mathcal{H}$, because the sample complexity only depends on the number of partner types.

2 METHOD

We take a two-player team Markov game as the collaboration environment. Player 1 is the AI agent we wish to train, and player 2 is the partner.

2.1 Ad Hoc Teamwork as Team Markov Game

A two-player team Markov game is defined as $(2, S, A, \mathbb{P}, H, K, r, \gamma)$, where S is the joint state space and A is the action space for both players. In this paper, we only consider the tabular setting, i.e.,

$|S| < \infty$ and $|A| < \infty$. The transition kernel $\mathbb{P}$ and the reward r are defined as: $\mathbb{P} : S \times A \times A \to \Delta(S)$, $r : S \times A \times A \to \Delta([0,1])$, where $\Delta(S)$ is the probability distribution over S and $\Delta([0,1])$ is the probability distribution over $[0,1]$. We also define the model $M\colon S \times A \times A \to \Delta(S \times [0,1])$ as:

$$M(s', R \mid s, a, b) \triangleq \mathbb{P}(s' \mid s, a, b) \cdot r(R \mid s, a, b).$$

The time horizon and the total number of episodes are denoted by H and K, respectively. The game is discounted by a discount factor $0 < \gamma \leq 1$. The term s_h^k denotes the joint state of episode k at time h. We use a_h^k and b_h^k to denote the actions of episode k at time h for player 1 and player 2, respectively. The immediate reward of episode k at time h is denoted as r_h^k. The initial state is fixed for all episodes and denoted as s_0, i.e., $s_0^k = s_0$ for all $k \in \{1, 2, \ldots, K\}$. The policy of player 1 is denoted as μ and defined as $\mu : S \times [H] \to \Delta(A)$, where $[H] = \{0, 1, \ldots, H-1\}$. The policy of player 2 is denoted as π and defined as $\pi : S \times [H] \to \Delta(A)$. The set of all policies available to player 1 is denoted by $\mathcal{U}$ and the set of all policies available to player 2 is denoted by Π. Now, we define the cumulative reward given policies (μ, π) as

$$\begin{aligned} V(\mu, \pi) = \mathbb{E}_{a_h \sim \mu(s_h, h), b_h \sim \pi(s_h, h)} & \\ & \left[\sum_{h=0}^{H-1} \gamma^h \mathbb{E}(r(s_h, a_h, b_h)) \right. \\ & \left. \mid s_0 = s_0, s_{h+1} \sim \mathbb{P}(s_h, a_h, b_h) \right] . \end{aligned}$$

We denote the set of potential partner policies as $\mathcal{H}^*$, which is a subset of Π. The AI agent collaborates across K episodes with a fixed yet unknown policy π^* from $\mathcal{H}^*$. However, the AI agent does not know $\mathcal{H}^*$ exactly. Instead, an online learning algorithm that solves the AHT problem is provided with a prior hypothesis set $\mathcal{H}$ that approximates $\mathcal{H}^*$. The algorithm that takes $\mathcal{H}$ K as input is denoted as **Alg**. The algorithm produces a series of AI agent policies $\{\mu^k\}_{k \in [K]}$, i.e., $\{\mu^k\}_{k \in [K]} = \mathbf{Alg}(K, \mathcal{H})$. We also use the term $V^*(\pi)$ to represent the optimal collaboration reward if the true policy is π, i.e., $V^*(\pi) \triangleq \max_{\mu \in \mathcal{U}} V(\mu, \pi)$. The regret function for an algorithm **Alg** is defined as

$$\mathrm{Reg}_{\mathbf{Alg}}(K, \mathcal{H}, \pi^*) = \sum_{k \in [K]} [V^*(\pi^*) - V(\mu^k, \pi^*)].$$

Therefore, the *goal of the ad hoc teamwork (AHT) problem* is defined as: "Generating a hypothesis set $\mathcal{H}$ and developing an algorithm **Alg** such that for all $\pi^* \in \mathcal{H}^*$, the regret $\mathrm{Reg}_{\mathbf{Alg}}(K, \mathcal{H}, \pi^*)$ is sublinear."

The potential partner policy set $\mathcal{H}^*$ may contain policies that, while distinct, lead to an identical optimal cumulative reward and best responses. This motivates us to categorize these policies into the same type. For any $\pi \in \Pi$, define the set of best response policies to π by $\mathrm{BR}(\pi) \triangleq \mathrm{argmax}_{\mu \in \mathcal{U}} \; V(\mu, \pi)$. We assume the existence of an oracle that can return a best response from $\mathrm{BR}(\pi)$.

Definition 2.1 (Best response oracle). A function $\psi\colon \Pi \to \mathcal{U}$ is called a *best response oracle* if $\psi(\pi) \in \mathrm{BR}(\pi)$ for any $\pi \in \Pi$.

The best response oracle is a choice function that selects one single AI agent policy within the set of best response policies. In this way, we can define the equivalence relation which only requires the values of their best response oracle to be the same.

Definition 2.2 (ψ-type). Two policies π and π' are said to be of *the same type under oracle* ψ (or simply *the same* ψ*-type*) if

$$\psi(\pi) = \psi(\pi') \quad \text{and} \quad V^*(\pi) = V^*(\pi').$$

The equivalence relationship is denoted as $\pi \overset{\psi}{\sim} \pi'$. When two policies π and π' are not of the same ψ-type, their relationship is denoted as $\pi \overset{\psi}{\nsim} \pi'$.

Following the definition of ψ-type, we introduce the concept of *type independence* for a hypothesis set.

Definition 2.3 (Type independence). A set Π of policies is called *type-independent* under oracle ψ if for all $\pi, \pi' \in \Pi$ such that $\pi \neq \pi'$, we have $\pi \overset{\psi}{\nsim} \pi'$.

Type independence gives rise to a characterization of the intrinsic complexity of a set of partner policies, called the *type number.*

Definition 2.4 (Type number). Given a policy set $\mathcal{H} \subseteq \Pi$, the type number $n^{\psi}(\mathcal{H})$ under oracle ψ is the size of any largest type-independent subset of $\mathcal{H}$.

Intuitively, the sample complexity of an online algorithm **Alg** increases as $|\mathcal{H}|$ increases. However, there exists an online algorithm whose sample complexity increases as $|n^{\psi}(\mathcal{H})|$ increases. Thus, the sample complexity is independent of $|\mathcal{H}|$.

2.2 Ignoring Similarity: Redundancy-Aware

Definition 2.5 (Redundancy awareness). We say an online algorithm is redundancy-aware if the regret of this algorithm depends only on $n^{\psi}(\mathcal{H})$ instead of $|\mathcal{H}|$. In other words, $\mathrm{Reg}_{\mathbf{Alg}}(K, \mathcal{H}, \pi) < \phi(n^{\psi}(\mathcal{H}))$ for some function ϕ.

Redundancy awareness implies that the sample complexity depends only on the type number.

Theorem 2.6. *The Maximize to Explore (MEX) algorithm developed in [5] is redundancy-aware.*

The above theorem shows the existence of a redundancy-aware algorithm.

3 EXPERIMENT

We benchmarked the MEX algorithm against several other algorithms, including the Upper Confidence Bound algorithm [4], Q-learning with UCB exploration [2, 3], and the optimistic posterior sampling algorithm [9]. Building on the approach in [8], we generated different types of partner policies to form a hypothesis set. The true partner policy was chosen from the hypothesis set. Our findings suggest that the MEX algorithm outperforms these algorithms on a large hypothesis set with a small type number. This demonstrates that the MEX algorithm is redundancy-aware.

4 CONCLUSION

We present the existence of a redundancy-aware algorithm, i.e., the sample complexity of this algorithm depends only on the number of types in the hypothesis set instead of the number of policies. Thus, eliminating the same type of policies from the hypothesis set is unnecessary if the learning algorithm is redundancy-aware.

REFERENCES

[1] Stefano V. Albrecht and Peter Stone. 2018. Autonomous Agents Modelling Other Agents: A Comprehensive Survey and Open Problems. *Artificial Intelligence* 258 (May 2018), 66–95. https://doi.org/10.1016/j.artint.2018.01.002

[2] Kefan Dong, Yuanhao Wang, Xiaoyu Chen, and Liwei Wang. 2019. Q-Learning with UCB Exploration Is Sample Efficient for Infinite-Horizon MDP. arXiv:1901.09311 [cs, stat]

[3] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I. Jordan. 2018. Is Q-learning Provably Efficient? https://doi.org/10.48550/arXiv.1807.03765 arXiv:1807.03765 [cs, math, stat]

[4] T. L Lai and Herbert Robbins. 1985. Asymptotically Efficient Adaptive Allocation Rules. *Advances in Applied Mathematics* 6, 1 (March 1985), 4–22. https://doi.org/10.1016/0196-8858(85)90002-8

[5] Zhihan Liu, Miao Lu, Wei Xiong, Han Zhong, Hao Hu, Shenao Zhang, Sirui Zheng, Zhuoran Yang, and Zhaoran Wang. 2023. One Objective to Rule Them All: A Maximization Objective Fusing Estimation and Planning for Exploration. https://doi.org/10.48550/arXiv.2305.18258 arXiv:2305.18258 [cs, math, stat]

[6] Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. 2022. A Survey of Ad Hoc Teamwork Research. In *Multi-Agent Systems*, Dorothea Baumeister and Jörg Rothe (Eds.). Vol. 13442. Springer International Publishing, Cham, 275–293. https://doi.org/10.1007/978-3-031-20614-6_16

[7] Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. 2010. Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. *AAAI* 24, 1 (July 2010), 1504–1509. https://doi.org/10.1609/aaai.v24i1.7529

[8] D. J. Strouse, Kevin R. McKee, Matt Botvinick, Edward Hughes, and Richard Everett. 2022. Collaborating with Humans without Human Data. https://doi.org/10.48550/arXiv.2110.08176 arXiv:2110.08176 [cs]

[9] Han Zhong, Wei Xiong, Sirui Zheng, Liwei Wang, Zhaoran Wang, Zhuoran Yang, and Tong Zhang. 2023. GEC: A Unified Framework for Interactive Decision Making in MDP, POMDP, and Beyond. arXiv:2211.01962 [cs, math, stat]