Learn a Robust Policy in Adversarial Games via Playing with an Expert Opponent

Extended Abstract

Jialian Li, Tongzheng Ren, Hang Su and Jun Zhu
Tsinghua University
Beijing, China
lijialia16@mails.tsinghua.edu.cn,rtz19970824@gmail.com,{suhangss,dcszj}@mail.tsinghua.edu.cn

ABSTRACT

Reinforcement learning methods such as AlphaZero have achieved super-human performance in adversarial games by training in a self-play manner. However, they generally require a large amount of computational resources to search for an (approximately) optimal policy in the joint state-action space involving both players and the environment. To accelerate the exploration process, we propose a new paradigm of "learning by playing" by considering the scenarios where expert opponents are accessible. By observing the opponent actions, the agent accelerates exploration by assigning more searching sources in these actions. To alleviate the sparse reward issue when facing the expert opponent at the beginning, we technically propose a novel method called Ladder Opponent Modeling (LOM), which builds a ladder opponent to facilitate the learning process. The agent plays with both the expert and ladder alternatively with its competence improved gradually. The online manner of the ladder opponent generates auxiliary tasks gradually, yielding a tractable improvement for the agent.

KEYWORDS

Reinforcement Learning; Extensive games; Sparse Reward

ACM Reference Format:
Jialian Li, Tongzheng Ren, Hang Su and Jun Zhu. 2019. Learn a Robust Policy in Adversarial Games via Playing with an Expert Opponent. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Two-player zero-sum games concern the tasks where two agents are involved and each tries to maximize its own reward. Since opposite rewards are given to each agent, an adversarial property exits. Many practical problems such as board games, competitive sports or some economic problems fall into this category. Our goal is to identify a policy for one of the player that performs well against any other opponent. In the adversarial two-player games, the target to find a policy that can gain good rewards universally requires taking the other player into consideration and searching in the joint state-action space. The joint space is much larger than that for a single agent and it is non-trivial to implement efficient exploration in such a space.

The state-of-the-art work AlphaZero [10] on board games works in a self-play manner. It trains policies for both players and let them play with each other until reaching well-performing policies. However, the searching in the joint policy space requires a huge amount of resources. For instance, AlphaZero used over 5000 TPUs to train board games. A potential way to accelerate the process is to learn the policy by mimicking a given expert player with imitation learning [7, 12], which is considered as "learning by watching". However, in practice, it may accumulate learning errors between the policies of experts and learners, resulting in poor performance especially in the scenario of sequential decision making.

In many practical scenarios, an expert opponent is accessible which allows our agent to play with it repeatedly. For example, a dealer in a casino and an attacker in network security [5] can be both considered as expert opponents. Notice that the expert opponent refers to an expert taking the role of the opponent, which is different from the expert in imitation learning. Intuitively, the given opponent can guide the agent to explore the states that are probable to meet in the face of other powerful opponents, making the agent improve its policy more efficiently. In this paper, we propose a new paradigm named "learning by playing", which encourages the agent to learn via playing with an accessible expert opponent. However, the key challenge for this paradigm is that is a beginner agent can hardly defeat the expert opponent, which makes it difficult for the agent to obtain meaningful rewards to improve its policy.

To address this challenge, we propose a novel method named Ladder Opponent Modeling (LOM). We introduce a ladder opponent to help the agent gain gradual improvement. In the training process, the agent updates its policy by playing with the expert opponent and the ladder opponent alternatively until convergence. When the agent plays with the expert opponent, the ladder opponent observes the behavior of the expert opponent and imitates the expert behavior. Since the ladder opponent is trained online, its policy is weak initially and gradually gets close to the opponent as more games are observed and played. When playing with the ladder opponent, the agent can gain positive rewards with higher probability, which alleviates the *sparse-reward* challenge.

Our work can be categorized from the perspective of "curriculum learning" [1, 4], which learns a set of increasingly more complicated auxiliary tasks gradually, yielding an effective performance in numerous scenarios. However, it is nontrivial to design the curriculum automatically, especially for the adversarial tasks. Our proposed LOM provides a method to generate the auxiliary tasks automatically which gradually becomes more complicated online.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

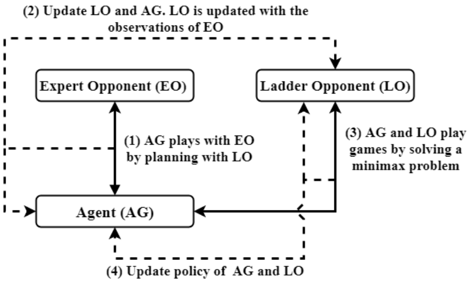
(2) Update LO and AG. LO is updated with the observations of EO
Expert Opponent (EO)
Ladder Opponent (LO)
(1) AG plays with EO by planning with LO
(3) AG and LO play games by solving a minimax problem
Agent (AG)
(4) Update policy of AG and LO

Figure 1: Configuration of training process for the proposed LOM method

2 PRELIMINARIES

Markov decision process (MDP) is a general framework in RL defined with a tuple $M = \langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where S and A respectively denote the state and action spaces, $\mathcal{P} : S \times A \times S \to \mathbb{R}$ denotes the state transition probability, $\mathcal{R} : S \times A \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1]$ is a discounting factor. A policy π maps every state $s \in S$ to a distribution over the action set A.

In a two-player zero-sum extensive game with perfect information [6], an agent is solving an MDP once its opponent uses a fixed policy. More formally, each player P_i ($i \in \{1, 2\}$) has its state space S_i, action space A_i and reward function $\mathcal{R}_i$. Specifically, $\mathcal{R}_1 = -\mathcal{R}_2$ and $\mathcal{R}_i(s) = 0$ unless s is a terminal state. Denote π_i as the policy for P_i and Δ_i as the set of all π_i. When Player $P_{\{1,2\}\setminus i}$ fixes its policy $\pi_{\{1,2\}\setminus i}$, P_i is solving an MDP $M_i = \langle S_i, A_i, \mathcal{P}_i, \mathcal{R}_i, \gamma \rangle$ where $\mathcal{P}_i = \pi_{\{1,2\}\setminus i}$ and $\gamma = 1$. Then we can define the expected reward for player i against the other player when their policy tuple is (π_1, π_2) as:

$$u_i(\pi_1, \pi_2) := \mathbb{E}_{\pi_1, \pi_2}\left[\sum_{t=0}^{T} \mathcal{R}_i(s_t)\right]. \tag{1}$$

We can train both agents to maximize their own rewards in the self-play paradigm [2, 10, 11]. Although they are probable to reach near-optimal policy, these methods are computationally inefficient because they need to explore the policy space for both players.

We can apply imitation learning to solve these games if supervised information is given [7, 12]. Imitation learning can accelerate learning, but it can result in error accumulation [3, 8, 9].

3 LADDER OPPONENT MODELING

In our "learning by playing" paradigm, we consider the game that an *expert opponent* is given. For convenience, our *agent* takes the role of P_1 and the given opponent takes P_2. Denote the policy of the opponent to be π^o. Intuitively, we aim to solve $\max_{\pi^a \in \Delta_1} u_1(\pi^a, \pi^o)$. The main problem for this problem formulation is that general RL methods are hard to improve since an expert π^o leads to sparse positive rewards. A further issue is that the solution are not ensured to perform well against other opponents.

To address the main issue, we introduce a *ladder opponent* to help the *agent* to improve. Since u_i is assumed to be bounded, $u_1(\pi_1, \pi_2)$ is a L-Lipschitz function on either π_1 or π_2, where L can be large. Thus for any $\pi_2' \in \Delta_2$, $|u_1(\pi_1, \pi_2) - u_1(\pi_1, \pi_2')| \le L * d(\pi_2, \pi_2')$. Then we have

$$u_1(\pi^a, \pi^o) \le u_1(\pi^a, \pi^\ell) + L * d(\pi^\ell, \pi^o). \tag{2}$$

This upper bound equals to the original function when $\pi^\ell = \pi^o$. Hence, we turn to optimize

$$\max_{\pi^a \in \Delta_1} \min_{\pi^\ell \in \Delta_2} u_1(\pi^a, \pi^\ell) + L * d(\pi^\ell, \pi^o). \tag{3}$$

Notice that an *expert opponent* should be powerful. The first term above is likely to decrease as π^ℓ gets close to π^o. Hence we change the second term into a constraint. Since we can only observe actions chosen from π^o, the distance can only be computed on the observed data. Then we give an approximation to problem (3)

$$\max_{\pi^a \in \Delta_1} \min_{\pi^\ell \in \Delta_2} u_1(\pi^a, \pi^\ell), \quad s.t.\ d(\pi^\ell, \pi^o)_{obs} < \delta, \tag{4}$$

where $d(\pi^\ell, \pi^o)_{obs}$ denotes the empirical divergence between π^ℓ and π^o on the observed part of π^o.

Optimizing problem (4) leads to our method Ladder Opponent Modeling (LOM). As shown in Fig. 1, the *agent* first plays with the *expert opponent*, and then the *ladder opponent* tries to satisfy the constraint with the observation data from π^o. Then the *agent* plays with the *ladder opponent* then to optimize the objective function. We give the algorithm in Alg. 1.

LOM can solve the sparse reward issue. The *ladder opponent* is weak at the beginning of training and the process that the *agent* plays with the *opponent model* and the *ladder opponent* alternatively can improve both the *agent* and the *ladder* gradually. As a byproduct, the games played by the *agent* and *ladder* can help the learned policy to be robust against other opponents.

Algorithm 1 Ladder Opponent Modeling

```
Initialization: Initialize $\pi^\ell$, $\pi^a$, $D_\ell = \emptyset$, $m_1$ and $m_2$.
repeat
    for $t = 1$ to $m_1$ do
        The agent uses $\pi^a$ and $\pi^\ell$ to search and uses the searching result to play with the expert opponent.
        For $(s, a)$ played by the expert opponent, $D_\ell = D_\ell \cup \{(s, a)\}$.
    end for
    $\pi^\ell \leftarrow \arg\min_{\pi_2 \in \Delta_2} d(\pi^o, \pi_2)_{D_\ell}$
    Update $\pi^a$ with the searching results
    for $t = 1$ to $m_2$ do
        The agent and the ladder opponent use $\pi^a$ and $\pi^\ell$ to search and use the searching results to choose actions.
    end for
    Update $\pi^a$ and $\pi^\ell$ with the searching results.
until Convergence
```

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (No.2017YFA0700904), NSFC projects (Nos. 61620106010, 61621136008, 61332007), the MIIT Grant of Int. Man. Comp. Stan (No. 2016ZXFB00001), Tsinghua Tiangong Institute for Intelligent Computing, the NVIDIA NVAIL Program and a Project from Siemens.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 41–48.

[2] George W Brown. 1951. Iterative solution of games by fictitious play. *Activity analysis of production and allocation* 13, 1 (1951), 374–376.

[3] He He, Jason Eisner, and Hal Daume. 2012. Imitation learning by coaching. In *Advances in Neural Information Processing Systems*. 3149–3157.

[4] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-Paced Curriculum Learning.. In *AAAI*, Vol. 2. 6.

[5] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. 2013. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)* 45, 3 (2013), 25.

[6] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic game theory*. Vol. 1. Cambridge University Press Cambridge.

[7] In-Seok Oh, Ho-Chul Cho, and Kyung-Joong Kim. 2014. Imitation learning for combat system in RTS games with application to starcraft. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*. IEEE, 1–2.

[8] Stéphane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 661–668.

[9] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 627–635.

[10] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv preprint arXiv:1712.01815* (2017).

[11] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

[12] Christian Thurau, Christian Bauckhage, and Gerhard Sagerer. 2004. Imitation learning at all levels of game-AI. In *Proceedings of the international conference on computer games, artificial intelligence, design and education*, Vol. 5.