Approximating Policy Space Repeating Oracle For Large Populations of Agents
Michael Krumdick, Geoffrey Liu

1 Introduction

Most multi-agent reinforcement learning (MARL) algorithms can be split up into two distinct steps. First, given our current population of agents, define the distribution of agents that minimizes exploitability using a meta-solver, such as regret minimization. Second, given this distribution, find a new agent that best exploits it, and add it to the population. This simple framework, called Policy Space Repeating Oracle (PSRO) [4], encompasses a wide range of different algorithms, including independent reinforcement learning, self-play, fictitious self-play, and iterated best response.

One of the core limitations of PSRO is the meta-solving step. Most meta-solvers are brittle and slow as the number of agents, N, increases. This effectively limits the maximum population size. What we want to develop is a scalable, stochastic approximation of one of the meta-solvers by framing the collection of payoffs between the agents as a multi-armed bandit problem. Imagine you have a large number of agents and an expensive evaluation function that computes the payoff for playing one agent against another. Effectively, what we want to find is an efficient algorithm for deciding which of those squares in our matrix to fill in. As M, our total evaluation budget, goes to N^2, we will have our exact solver from before. Intuitively, if we have a large number of agents, there exists a subset of those agents we can use to find a good approximate meta-solution. The hope is that by increasing the total agents to consider, finding a good subset should become easier.

2 Motivation

For a few decades, the large majority of research being done in reinforcement learning only considered the single-agent or independent learning case. Here, the environment is static. Anything being simulated as part of the environment will never learn how to adapt to the agent. For example, the common set of Atari baselines [3] include games that simulate other agents, i.e. the other paddle in pong, but these agents will never adapt in response to the agent. Multi-agent reinforcement learning (MARL) is any type of environment in which an agent has to learn an optimal behavior in the presence of other agents who are trying to optimize their own behaviors.

MARL has received large amounts of interest recently, as many people have identified it as a major gap for the practical use of reinforcement learning. Real world environment often involve dealing with other agents who are non-static. These approaches broadly have application areas including war gaming, advertising, trading, and policy simulation.

3 Plan

Our plan is to follow in a framework similar to [2]. We will first:

1. **Exploration:** Implement and analyze baselines in simple normal-form games. This gives a quick iterative feedback loop that we can use to compare our approximate algorithm to the ground truth solution.
2. **Design:** Through the lens of multi-armed bandit problems, attempt to design a scalable algorithm for computing the approximate equilibria.
3. **Application:** Modify this algorithm to work with deep reinforcement learning agents and demonstrate its application on a more complex environment.

Since this is a research project, we want to make sure that we have a good number of off-ramps when things do not go as planned. Once we have implemented our baselines, if we cannot develop an improved

algorithm, we can pivot to an application setup where we apply some of the mutli-agent algorithms to an interesting problem. The only step that is mandatory is the completion of our exploratory setup.

In terms of environments, we want to find something that is scalable. Meaning, we want a problem that can both work as a normal-form game (or similar) and by updating the parameters it can scale to something that will require deep reinforcement learning. By having a parameter or set of parameters that allow us to smoothly scale complexity, we can treat this as another dimension in our analysis. Meaning instead of having a sliding scale of separate environments that represent different levels of complexity, we can plot complexity as a function of a specific parameter. One possible environment that we current have in mind are Erdos-Selfridge-Spencer(ESS)[6] games. These not only fit the previous description, they also have an analytical optimal solution that can always be computed as a linear function of the state. In other words, we can compute the exploitability of an agent analytically without relying on an estimate. However, the main types of ESS games seem to have action spaces that would complicated to simulate as a simple normal-form game and they are lacking open source implementations.

4 Related Works

There are a few works we would like to build off of. The AlphaRank paper proposed a polynomial time solution to the equilibrium points for multi-agent systems. [5] However, this algorithm has been shown to be time consuming to compute in practice due to the exponentially large payoff matrix. Other works have tried to improve it's performance, such as [1], but the majority of these try to speed up the calculation of the equilibrium given the large matrix. What we want to do is instead try to use our collection strategy to convert the possibly large matrix into a smaller matrix with a solution of approximately equal quality.

References

[1] α^{α}-rank: Scalable multi-agent evaluation through evolution. *CoRR*, abs/1909.11628, 2019. Withdrawn.

[2] Thomas Anthony, Tom Eccles, Andrea Tacchetti, János Kramár, Ian Gemp, Thomas C. Hudson, Nicolas Porcel, Marc Lanctot, Julien Pérolat, Richard Everett, Roman Werpachowski, Satinder Singh, Thore Graepel, and Yoram Bachrach. Learning to play no-press diplomacy with best response policy iteration, 2020.

[3] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012.

[4] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning, 2017.

[5] Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M. Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. alpha-rank: Multi-agent evaluation by evolution, 2019.

[6] Maithra Raghu, Alex Irpan, Jacob Andreas, Robert Kleinberg, Quoc V. Le, and Jon Kleinberg. Can deep reinforcement learning solve erdos-selfridge-spencer games?, 2018.