Multi-agent RL for Cooperation in Social Dilemmas

Jared Weinstein
Advisor: Marynel Vázquez
https://github.com/social-dilemma/multiagent

1 ABSTRACT

Reinforcement learning has proven to be extremely successful at getting a single agent to solve a single goal. In real life, however, tasks often require the joint cooperation of many actors with different intentions, information, and abilities. Multi-agent systems introduce a significant degree of complexity. Recent work on multi-agent reinforcement learning (MARL) focuses on social dilemmas. In a social dilemma, the interests of an individual conflict with that of the group. Humans adeptly and intuitively solve social dilemmas by engaging in complex, temporally extended tasks like communication, punishment, and action prediction effectively working together towards a common goal while preventing free-riders from taking advantage of free work. Because social dilemmas require these complex interactions, they are a perfect setting to explore techniques in multi-agent reinforcement learning. Under these conditions, is it possible for artificial agents to learn the behaviors necessary to act in a prosocial manner?

My work represents an initial step in exploring the challenges of multi-agent reinforcement learning in a social dilemma. I start by building a gridworld version of the classic prisoner's dilemma and use the Asynchronous Actor-Critic Agents (A3C) algorithm to simultaneously learn the optimal policy for two agents. The training of independent agents with mutually dependent policies raises an issue of stability. I explore this issue along with the impact of discount rate, entropy, and fixed agent policies on cooperation. In the end, a hyperparameter search successfully results in fully cooperative behavior between two trained agents. However, the learned policy is naive and easily exploited by a human.

2 INTRODUCTION

Given the success of reinforcement learning, artificially intelligent systems are likely to become increasingly integrated into complex real-world environments. These systems often demand strong social coordination. In the past, centralized training has been used to effectively coordinate multiple individual actors working towards the same shared goal. However, centralized control is not always possible nor desirable. Autonomous navigation, for example, is unlikely to be solved by central coordination. Instead, multiple trained intelligent actors will choose actions based only on local information. In these situations, how do we ensure that multiple actors motivated by different (and often contradictory) goals, work together in a harmonious manner?

Social dilemmas provide a complex environment in which the interests of an individual agent conflict with the collective interest. Leibo et al. outlines three canonical two-player social dilemmas in matrix form: Chicken, Stag Hunt, and Prisoner's Dilemma. [2] In each game, agents make simultaneous decisions to either defect or cooperate. Games are played repeatedly and rewards are distributed and accumulate over time. Figure 1 shows the payoff matrix for a single round of each variation.

Chicken	C	D
C	3, 3	1, 4
D	4, 1	0, 0

Stag Hunt	C	D
C	4, 4	0, 3
D	3, 0	1, 1

Prisoners	C	D
C	3, 3	0, 4
D	4, 0	1, 1

Figure 1: ***Payoff matrices for common forms of a two-player simultaneous social dilemma.***

Each of these games fulfills four conditions to make it a social dilemma. [2]

1. *Mutual cooperation is better than mutual defection*
2. *Mutual cooperation is better than cooperation and defection*
3. *Mutual cooperation is preferred to being exploited by a defector*
4. *Exploiting a cooperator is preferred over mutual cooperation*

My work focuses primarily on a variation of the Prisoner's Dilemma (PD). In a single instance of the game, defection is always the logical choice. Regardless of the other player's policy, defecting results in a higher reward. However, when the game is played repeatedly, the strategy changes. Opponents can choose to punish defection in future rounds. Although this retributive action lowers group reward, it motivates cooperation. Defect once and you risk getting punished for many rounds into the future.

Generous tit-for-tat is one such common strategy that punishes defection and incentivizes cooperation. The strategy is simple: for each round, copy the opponent's action from the previous round. Alexander Peysakhovich and Adam Lerer note that generous tit-for-tat strategy is appealing because it meets certain desirable properties. It begins by cooperating, is easily understood, cooperates with cooperators, doesn't get exploited by defectors, and returns to cooperative equilibrium. [3] For these reasons, generous tit-for-tat offers a fairly robust solution to the prisoner's dilemma. Unfortunately, it makes heavy-handed assumptions about the environment and does not generalize to more complex situations; however, I keep these positive attributes in mind as I evaluate my trained policies.

3 RELATED WORKS

My work is part of a recent explosion of research in Multi-agent Reinforcement Learning (MARL). I will discuss recent formulations of both the environment and training method.

3.1 Environment

Two player matrix games (like Chicken, Stag Hunt, and Prisoners discussed above) are the simplest formulation of a social dilemma. Additional complexities can be added while preserving the fundamental conflict between individual and group goals.

Kleiman-Weiner et al. introduce simple *gridworld* games. Rather than a binary choice, cooperation and defection become actions across both space and time that require low-level planning over spatial actions to realize long-term strategic goals. The action space is much larger than a simple matrix game and cooperative strategies are less obvious. [8] Leibo et al. build on the gridworld environment by creating two two-player *partially observable* Markov Games: Gathering and Wolfpack. The games consist of a discrete set of states, an observation function that maps states to a limited view for each agent, and a set of allowable actions. The actions of each agent and the prior game state determine the subsequent state of the game. Rewards are distributed individually to each agent based on the current state. [2] This formulation of the social dilemma is complex enough to allow a huge variety of different games.

Jaques et al. further complicates the environment by extending it beyond two-player games. In their work, *numerous actors* must work together simultaneously. [7]

Additions to the number of game states, limiting the view of each agent, and increasing the number of agents all challenge our ability to accomplish human level cooperative behavior. As the environment diverges from its initial humble formulation as a discrete matrix choice, more intense learning mechanisms are required to learn successful behavior.

3.2 Training

The common approach to learning is a deep Q-network, a technique used by Leibo et al. Their network maintains a strong condition of independence between agents; weights are not shared and each agent is trained independently on unique rewards. Other agents appear only as part of the environment. As an unfortunate consequence of this independence, the environment becomes non-stationary. As agents adapt their strategy over time, the environment of other agents appears to be changing, and the learning target moves. I discuss difficulties related to the non-stationary nature of MARL in more detail in the results section. [2] [11]

Jaques et al. and Hughes et al. both break this strong condition of independence in order to provide a more powerful reward function.

Jaques et al. rewards causal influence. Agents are rewarded for actions that strongly impact the decisions of other agents. Jaques et al. argues that this reward is a form of social empowerment. In order to reward causal influence, agents must be able to recursively reason about what other agents would have done. [7] As such, they require direct access to the policy networks of other agents.

Hughes et al. modifies the reward function to punish inequity between agents. Inequity aversion is motivated by our human inclination towards fairness. Experiments with inequity-averse agents find that this modified reward function improves the chances of cooperative behavior over time. [5]

4 METHOD

My work extends primarily from the work of Leibo et al. in maintaining the independence condition for each trained agent. In addition, I use a similar formulation of the environment: a two-player, gridworld Markov game. I diverge from Leibo et al and instead rely on the network structure and hyperparameter values provided by Hughes et al. [2] [5]

The final implementation consists of three primary components. Pycolab provides an abstraction for the gridworld environment. It prevents agents from performing invalid actions, distributes rewards, and updates the game-state at each time step. RLLib is used extensively during training. It enables reinforcement learning using the A3C algorithm and does simple hyperparameter tuning. Finally, user interaction and game visualization is handled by Pygame. I'll discuss each component in further detail.

4.1 Environment

I implement a gridworld variant of the prisoner's dilemma. The game works by letting two agents, Red and Green, move freely along a line. At each step of the game, both agents independently choose to move left, move right, or stay still. The right side of the board represents cooperation while the left represents defection.

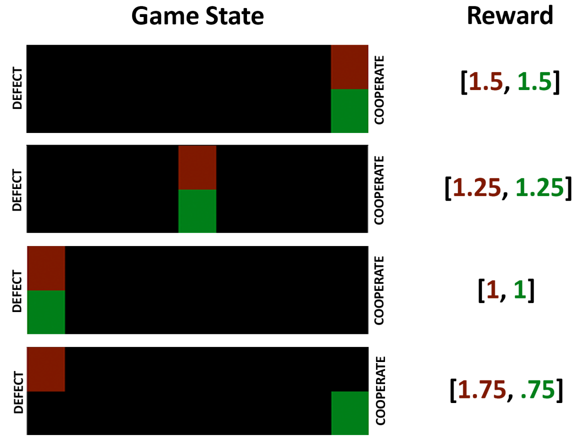

Figure 2: ***Rewards of 4 key situations during a game. Notice the best group reward results from mutual cooperation. However, the best individual reward occurs when Red defects and Green cooperates.***

Based on position, each agent A_i is assigned a cooperation value C_i and a defect value D_i. These values are bounded between 0 and 1 and maintain the condition that $C_i + D_i = 1$. They represent the degree to which an agent is cooperating and defecting. If an agent is far to the right (cooperating), $C_i = 1$ and $D_i = 0$. If an agent is in the middle, $C_i = .5 = D_i$. If an agent is flush left (defecting), $C_i = 0$ and $D_i = 1$ All values in the middle are linearly distributed.

Every n steps of the game, the reward is distributed based on each agent's position. I chose $n = 10$ to allow both agents sufficient time to move to any new position on the grid prior to the next reward. The reward for agent j interacting with n other agents is defined according to the function:

$$R_j = \alpha \sum_{i=1}^{n} C_i + D_j \tag{1}$$

In the two player case, the four conditions of a social dilemma are upheld if $0.5 > \alpha > 1$. [9] Notice that all agents benefit from every cooperative component; but, because $\alpha < 1$, a switch to defection increases individual reward. Figure 2 shows the reward values calculated with $\alpha = .75$. A quick check demonstrates that all conditions are properly met.

1. *Mutual cooperation is better than mutual defection*
   $1.5 + 1.5 > 1 + 1$
2. *Mutual cooperation is better than cooperation and defection*
   $1.5 + 1.5 > 1.75 + .75$
3. *Mutual cooperation is preferred to being exploited*
   $1.5 > .75$
4. *Exploiting a cooperator is preferred to mutual cooperation*
   $1.75 > 1.5$

Since these conditions hold at the extremes and intermediate values are calculated linearly, the game is a social dilemma.

This environment is much simpler than the social dilemmas implemented by Hughes et al. and Leibo et al allowing policies to train quickly and be easily understood. Yet, there are still complexities that make the problem interesting. For instance, cooperation and defection are temporally extended actions. A single movement to the left or right might not represent an overall commitment to cooperate or defect. In addition, because rewards are distributed every n time steps, the actions right before a reward is distributed become more significant than those immediately after a reward is distributed. Although these complexities are obvious to human players, it remains to be seen whether artificial intelligence will react in a similar manner.

4.2 Training

The first layer of my network is a single 2d-convolution layer with kernel size = 3 and stride = 1. The convolutional layer is connected to two fully connected layers ($n = 32$). The size of the output ($n = 3$) is determined by the action space of each agent. In this case, agents can move right, left, or stay put.

I use the A3C algorithm (Asynchronous Advantage Actor-Critic) for reinforcement learning. RLLib provides a simple implementation. [10]

In order to validate my training method, I experimented with manually fixing one agent's policy to simplify the problem from MARL to classic RL. Agent$_0$ is manually assigned one of three policies: full cooperation always moves to the right, full defection always moves to the left, and tit-for-tat copies the opponent's action at the previous time step (A_1^{t-1}).

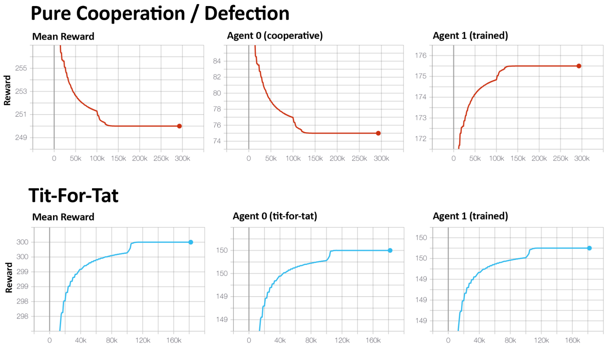

Figure 3: As expected, only when playing against tit-for-tat is cooperative behavior learned.

Figure 3 shows the average and individual rewards as training progresses. In both the fixed cooperation and fixed defection cases, the agent learns to always defect. This intuitively makes sense because there is no future punishment for non-cooperative behavior. On the other hand, when agent$_0$ has a fixed tit-for-tat strategy, agent$_1$ learns that non-cooperative actions are punished in the future. As a result, both agents cooperate fully and the mean reward is maximized.

5 RESULTS

To evaluate the effectiveness of my network, I conduct a small matrix search over the discount rates $\gamma = [.9, .99, .999]$ and entropy values $\epsilon = [.05, 0.005, 0.0005, .00005]$.

I expect a higher discount rate to train cooperative behavior. On the other hand for a myopic agent, the environment becomes a one-shot version of the prisoner's dilemma and defection is always the preferred choice. This correlation was observed for three of four fixed epsilon values. Figure 4 shows the positive correlation when $\epsilon = .005$. More work is required to determine whether the observed correlation between cooperation and discount rate is statistically significant.

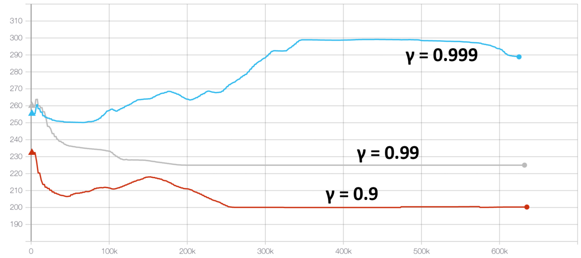

Figure 4: Agents concerned about the future behave more cooperatively. $\epsilon = 0.005$

The entropy value has a more complex relationship with cooperative behavior because it represents a trade-off between exploration and exploitation. Depending on the initial weights of network parameters, a high entropy value might be beneficial or harmful. In experiments with a fixed gamma value, I observe little to no correlation between entropy and overall cooperative behavior.

Overall, $\gamma = 0.999$, $\epsilon = 0.0005$ resulted in near perfect cooperative behavior for both agents. Although these results are promising, they fell short when tested against a human player. Both agents learned a naive policy of always cooperating, regardless of the opponent's actions. This naive policy is easily exploitable by a human player that quickly learns that defection incurs zero repercussions.[*]

In addition, the training is highly unstable. Even with the most successful hyperparameters, at the end of training, Agent$_0$ was beginning to switch to a more defective policy. Figure 5 shows another training period which was particularly volatile. Stability represents the biggest hurdle for MARL. The work of Hernandez-Leal et al.

[*] Visualizations of this trained behavior: https://youtu.be/FcQkoinrXdg

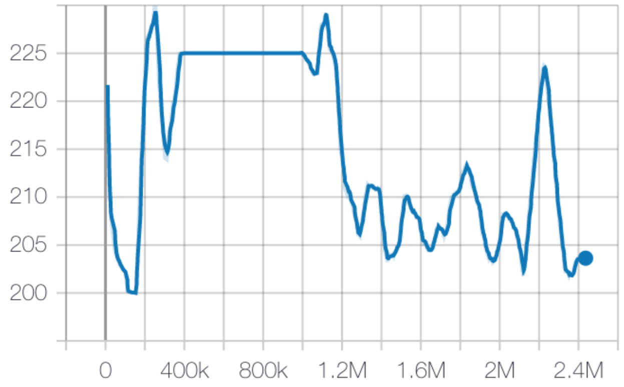

Figure 5: ***Instability in reward over 2000 training iterations.***

provides a comprehensive look at the issue of non-stationary environmental factors in MARL (also known as the moving target problem). They note that in multi-agent situations, each agent must learn the dynamics of the environment before adapting behavior to a target objective. However, because agents are learning simultaneously, the environment dynamics are constantly changing during the learning process. As a result, there is no theoretical guarantee of convergence towards a solution like in classic RL. Hernandez-Leal et al. presents several methods for addressing a non-stationary environment. [11] However, implementing these changes is beyond the scope of my project.

6 FUTURE WORK

This work represents only a preliminary exploration into the challenges of multi-agent reinforcement learning. Possible avenues for future work include, but are not limited to, expanding the hyperparameter search, modifying the α and n values of the environment, using non-linear rewards, creating new environments, introducing partial-observations, allowing communication between agents, and modifying the neural network architecture. There are also larger problems that demand attention. How can training be modified to prevent the moving-target problem while maintaining the condition of independence between agents? What modifications to the reward function will result in better cooperative behavior? Can agents with non-identical network architectures cooperate more effectively?

In addition, this work focused primarily on the interaction between two artificial agents. What happens when a human is introduced to the system? This adds another layer of complexity. How can we train a policy that adapts in real-time to different human players? Do human players act differently when playing against artificial agents? How does the behavior of artificial agents in a social dilemma impact how they are perceived by human observers? I hope these questions serve to excite and motivate future work in MARL. There is still a tremendous amount to be learned.

REFERENCES

[1] Wedekind, Claus, and Manfred Milinski. "Human Cooperation in the Simultaneous and the Alternating Prisoner's Dilemma: Pavlov versus Generous Tit-for-Tat." *Proceedings of the National Academy of Sciences*, vol. 93, no. 7, 1996, pp. 2686–2689., doi:10.1073/pnas.93.7.2686.

[2] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multiagent Reinforcement Learning in Sequential Social Dilemmas. *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp 464–473, 2017.

[3] Lazaridou, Angeliki, Alexander Peysakhovich, and Marco Baroni. "Multi-agent cooperation and the emergence of (natural) language." *arXiv preprint arXiv:1612.07182* (2016).

[4] Sally, D. 1995. Conversation and cooperation in social dilemmas: a meta-analysis of experiments from 1958 to 1992. *Ration. Soc.* 7(1):58–92

[5] Hughes, Edward, et al. "Inequity aversion improves cooperation in intertemporal social dilemmas." *Advances in Neural Information Processing Systems.* 2018.

[6] Yang, Jiachen, et al. "CM3: Cooperative Multi-goal Multi-stage Multi-agent Reinforcement Learning." *arXiv preprint arXiv*:1809.05188 (2018).

[7] Jaques, Natasha, et al. "Intrinsic Social Motivation via Causal Influence in Multi-Agent RL." *arXiv preprint arXiv*:1810.08647 (2018).

[8] Kleiman-Weiner, Max, et al. "Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction." *COGSCI.* 2016.

[9] Le, Stephen, and Robert Boyd. "Evolutionary dynamics of the continuous iterated prisoner's dilemma." *Journal of theoretical biology* 245.2 (2007): 258-267.

[10] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." *International conference on machine learning*. 2016.

[11] Hernandez-Leal, Pablo, et al. "A survey of learning in multiagent environments: Dealing with non-stationarity." *arXiv preprint arXiv:1707.09183* (2017).