A Study of Reinforcement Learning in the Iterated Prisoner's Dilemma

Jiyao Li
Department of Computer Science
Utah State University
Logan, Utah 84322
Email: jiyao.li@aggiemail.usu.edu

***Abstract*—Reinforcement Learning (RL) is widely used in unknown environments where agents impossibly learn from training sets. This paper is an empirical study of RL in the Iterated Prisoner's Dilemma (IPD), where defect is no longer dominant strategy and agent is expected to learn by itself to adapt to the change of the environment. Using RL algorithm become more challenge in such a complicated domain. The purpose of this paper is to investigate the ability of three different types of RL Agents which are implemented by Adaptive Dynamic Programming(ADP), Temporal-Difference Learning (TD) and Monte Carlo Method (MC) respectively. On the one hand, we hope that such RL Agents may adjust its strategy to do the best response when it competes with a Fixed Strategy Agent; on the other hand, RL Agent is also expected to elicit cooperation when it plays with itself or other types of RL Agents, which is supposed to be the best way to achieve more payoff for itself.**

Key Words—Reinforcement learning, machine learning, multiagent learning, adaptive dynamic programming, temporal-difference learning, Monte Carlo Method, prisoner's dilemma, game theory.

I. INTRODUCTION

Reinforcement Learning (RL) [11] is based on the idea that the tendency to produce an action should be strengthened (reinforced) if it produces favorable results and weakened if it produces unfavorable results in complex and dynamic environment. One of the advantages of RL is that, unlike supervised learning that require a set of training examples in terms of input-output pairs, agents can learn the strategy function by itself through the reward and punish signal from the environment. Another advantage is that RL algorithm uses Markov Decision Process (MDP) [7] as mathematical framework for modeling decision making process. Therefore, it is well-suited for use in the repeated games against an unknown opponent.

The Prisoner's Dilemma (PD) is a standard example of a matrix game in game theory. It shows that both rational agents would choose defect as their dominant strategy even though both cooperation may maximize the social welfare. However, in the Iterated Prisoner's Dilemma (IPD) [1] where the purpose of agents is to maximize its payoff through out the iterated games, there is a paradox that certain action may bring high immediate reward value for the agent whereas it would be harmful for the long term reward value. Moreover, the opponent is unknown so that learning agents have no any prior knowledge at all. Therefore, designing strategy scheme coping with such stochastic environment is very difficult.

The remainder of the paper is organized as follows. Section 2 provides some background information, which would introduce the rules of the IPD and some related works on it. Section 3 describes the learning agents in more detail and the following section describes the experiments. The last two sections contains conclusions and suggestions for future research.

II. BACKGROUND

A. *Rules of Iterated Prisoner's Dilemma (IPD)*

Both agents in the Prisoner's Dilemma (PD) game may choose Defect (D) or Cooperate (C) according to their own interests. Below are the rules of PD games:

- If both agents choose D, then they both will only earn 1 utility respectively.
- If one agent chooses D but the other selects C, the agent who defects may earns 5 utilities while the agent who cooperates gets nothing.
- If both agents choose C, then they both will earn 3 utilities respectively.

In IPD, two agents play Prisoner's Dilemma more than once in succession and they can remember previous actions of their opponent and choose their strategy accordingly, but they do not exactly know when the iterated game will be over. Both agents are trying to maximize their own utility.

B. *Related Works*

Most of research works seem to cover in the field of Q-learning. Sandholm and Crites [9] implement a Q-learning agent with lookup table and recurrent net respectively, their experimental results show that Q-learning agent can adjust strategy to a fixed strategy player but they seldom converge to cooperate when Q-learning agents play with each other. Babe [2] ever tried to use Social Reward Shaping [6] to improve the performance on cooperation, but this method can only accelerate the speed of convergence of RL but not guarantee that the outcome converges to cooperation. Moriyama proposed the *utility-based Q-learning concept* [3] [4] to improve the cooperation rate on Q-learning agents, he also discovered the relationship of payoffs that can guarantee the Q-learning agent to converge to cooperation [5]. Shibusawa proposed a novel approach called *expectation of cooperation strategy* [10] to guarantee cooperation in the Q-learning agent network.

However, both Moriyama and Shibusawa did not demonstrate that their Q-learning agents may both adjust their strategy when they compete with Fixed Strategy Agent and may also converge to cooperation when they play with other types of learning agents.

III. METHOD

Reinforcement learning is learning how to map states to actions so as to maximize the reward signal from the environment. As Figure 1 shown, RL Agent perceives Reward and State value as well as constructs its own internal state by reshaping the external state from environment. Then the internal state and the reward value are used to update the value of Q table by three various types of methods: Adaptive Dynamic Programming (ADP), Temporal Difference Learning (TD) and Monte Carlo Method (MC) respectively. In the Action Selection module, *soft-max* function is applied to determine whether to exploit and explore, and the appropriate action would be chosen and then sent back to the environment.

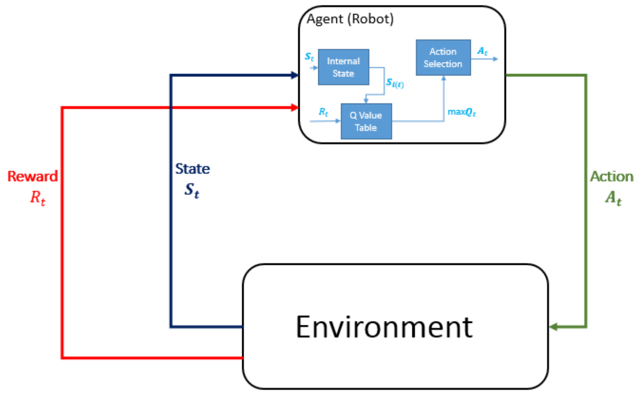


Fig. 1: The framework of Reinforcement Learning.

In the internal state module, if the state is considered to be the entire history, the internal states would increase in dimension with respect to the state at the previous decision point so that each state is visited at most once. Therefore the theoretical convergence result of RL do not apply. We assume that the IPD as a Markov Decision Process (MDP) where the situation of the next state is only determined by that of the current state, regardless of all other history state. So the state is viewed as previous actions of the agent and its opponent.

Q-Value is a value of state-action pair which indicate the score of action in certain state. Through the Q-Value table, the best action $\max_a Q(a,s)$ can be selected. The Q-Value table is updated by three different approaches: DP, TD and MC separately. The description of the methods is shown below:

- The Adaptive Dynamic Programming (ADP) [8] is similar to the Dynamic Programming (DP) where it uses the best future action-state value estimation $Q(a^{'},s^{'})$ and immediate reward $R(S)$ to evaluate the current action-state value $Q(a,s)$ through the transitional function. In the IPD games, agents do not know the transitional function of the environment so that we use statistical method to approximate it, so this is why it is called ADP. The Bellman Equation that demonstrate the DP is shown below:

$$Q(a,s) = R(S) + \gamma \sum_{s^{'}} T(s,a,s^{'}) \max_{a^{'}} Q(a^{'},s^{'})$$

  where γ is a discount rate $0 < \gamma < 1$, T is the transitional function.

- The Temporal Difference Learning (TD) [11] uses the present knowledge $Q(a^{'},s^{'})$ and immediate reward $R(s)$ to update the past experience $Q(a,s)$, which can be used to choose action in future. The advantage of TD is that it may avoid the transitional function of the environment. The update equation for TD is below:

$$Q(a,s) \leftarrow Q(a,s) + \alpha \left(R(s) + \gamma \max_{a} Q(a^{'},s^{'}) - Q(a,s) \right)$$

  Where α is learning rate $0 < \alpha < 1$ and γ is discount rate $0 < \gamma < 1$

- The Monte Carlo Method (MC) [11] also does not need to know the transitional model of the environment. It learns knowledge (the Q values)from the samples of each episodes, and uses such knowledge to choose its action. In the IPD, we calculate 10 samples of each episodes, sum up the score of each state-action value and then average it.

In Action Selection, it is not sufficient for RL Agent to always choose the best action from the Q-Value table since it would easily tend to be local optimal, it should try other actions as well to identify the environment. In this paper, Boltzmann Distribution is chosen as the soft-max function, the length of the history is used to decreased the temperature. Specifically, RL Agent's probability of selecting action a_i from state s is

$$p(a_i) = \frac{e^{Q(s,a_i)/t}}{\sum\limits_{a \in C,D} e^{Q(s,a)/t}}$$

where the temperature t is a function of the number of n of PD games so far:

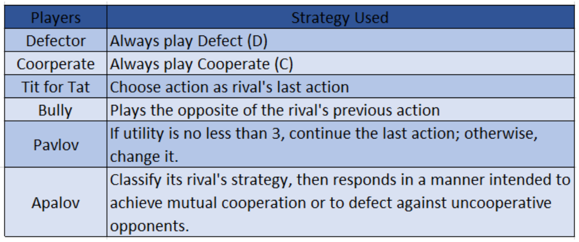
$$t = 20 \cdot 0.999^n$$

If $t < 0.1$ then no exploration is performed, the action with the highest Q value will be chosen with certainty. These constant for the annealing schedule (20,0.999,0.1) were chosen experimentally.

IV. RESULTS

In all of the experiments below, TDBot, DPBot and MCBot represents RL Agents implemented by TD, ADP and MC separately. The learning rate of the TDBot is decreasing as the iteration number goes up. The discount rate of all RL agents is set as 0.9. All other parameters are set as the Method section mentioned.

First of all, 3 tournaments that use Robin-Round style are held to test each RL Agent (Robot) playing with 6 other Fixed Strategy Agents, as shown in Figure 2. In the tournament, each

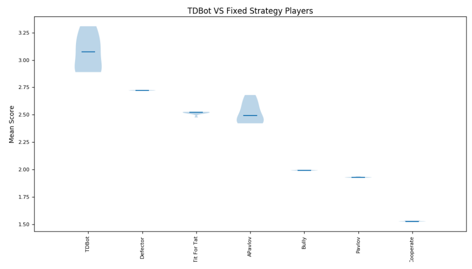


Players	Strategy Used
Defector	Always play Defect (D)
Coorperate	Always play Cooperate (C)
Tit for Tat	Choose action as rival's last action
Bully	Plays the opposite of the rival's previous action
Pavlov	If utility is no less than 3, continue the last action; otherwise, change it.
Apalov	Classify its rival's strategy, then responds in a manner intended to achieve mutual cooperation or to defect against uncooperative opponents.

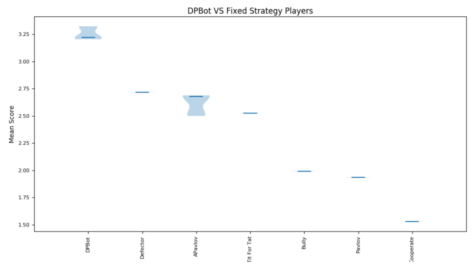
Fig. 2: Strategies used for variety of Fixed Strategy Players.

of the two agents play 10000 rounds in one match and the Robin-Round repeats 30 times.

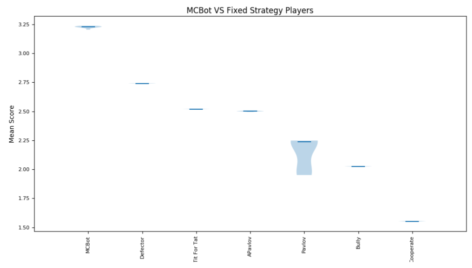
Figure 3 shows RL agents' mean score and its distribution competing with other 6 Fixed Strategy Agents in tournament. We can see that all RL Agents work better than the Fixed Strategy Agents. The mean score of DPBot and MCBot is 3.25, specially DPBot's score distribution seems to upper than its mean score while MCBot's distribution is focus on its mean score, as (b) (c) shown. For TDBot, its distribution range is so large, from 3.23 to 3.3, as (a) shown.



(a)



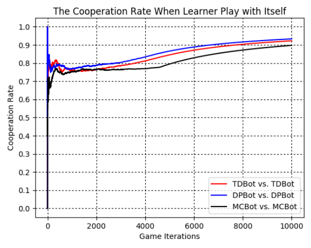
(b)



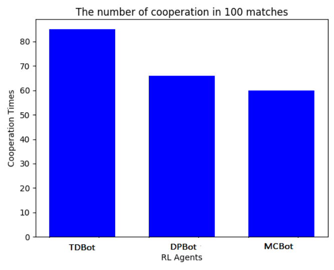
(c)

Fig. 3: The mean score and its distribution when RL Agents play with other 6 Fixed Strategy Agents in tournament.

Secondly, the homogeneous RL Agents matches are held. Here homogeneous means the agent plays with itself, so matches are TDBot vs. TDBot, DPBot vs. DPBot and MCBot vs. MCBot. Every two of the agent play 100 matches that contain 10000 rounds. Figure 4 (a) the cooperation rate would be above 90% when RL Agents can converge to cooperation. (b) TDBot's cooperation times are more than DPBot's and MCBot's, which are about 85%, 66% and 60% separately.



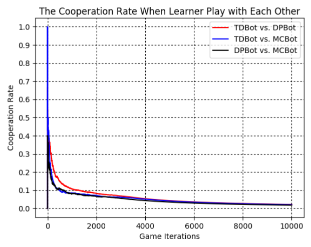
(a)



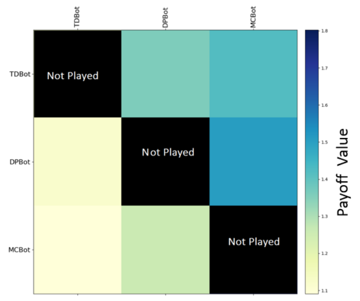
(b)

Fig. 4: (a) The trend of cooperation rate in the match that converge to cooperation (b)Cooperation times of each RL Agents who may elicit cooperation during the 100

At last, heterogeneous RL Agents' matches are held. Heterogeneous here means agents implement their RL with different methods. So matches are TDBot vs. DPBot, TDBot vs. MCBot and DPBot vs MCBot. Nearly no chance of cooperation would occur between heterogeneous RL Agents. Also, tournament that contains 10000 rounds in one match and the Robin-Round repeats 30 times is held among different types of RL Agents. Figure 5 (a) the cooperation rate of every RL Agents in each match closes to zero (b) TDBot seems to be stronger than the other two RL Agents while MCBot seems to be the weakest.



(a)



(b)

Fig. 5: (a)The Cooperation Rate When Learner Plays with Each other (b) The tournament between heterogeneous RL Agents

V. CONCLUSION

All three types of RL agents learned to play optimally against the Fixed Strategy Agents. Playing against Fixed Strategy Agents is analogous to learning in a single agent environment, because Fixed Strategy Agents does not learn: its mapping from internal state to action is fixed. Moreover, Cooperator and Defector ignore the state and choose one fixed action, Tit For Tat, Bully and Pavlov uses only one previous move as its internal state, Apavlov only classified its rival in 4 fixed catalog even though it would consider previous 6 moves. Therefore, learning against Fixed Strategy Agent is equivalent to learning in an environment where the true state does not increase in dimension and can be revisited.

Cooperation could be elicited when the same type of RL agent plays with itself. This is mainly because they share the same learning mechanism so that it is possible to converge in cooperation state. The cooperation rate of DPBot and are lower than TDBot because DPBot's Q-Value estimation is based on the transitional function that is only approximate the true environment, and MCBot's Q-Value evaluation is calculated through the samples of each episode that are not connected to each other.Surprisingly, the cooperation rate of TDBot increases when smaller learning rate is used, whereas it may not adjust its strategy well when playing with Fixed Strategy Agent in such situation.

However, cooperation seldom emerged in experiments when different types of RL agents play with each other. In such situation, the environment become non-stationary because agents use a different decision scheme and they may choose actions stochastically. Although agents could learn optimal Q-Value even in stochastic domains, the convergence proof dose not apply in non-stationary settings.

In the tournament among heterogeneous RL Ageants, DPBot and MCBot are apparently weaker than TDBot. The reason might be the transitional model of DPBot cannot predict the true environment precisely in a non-stationary environment and the samples of MCBot are relatively isolated in episode.

VI. FUTURE RESEARCH

For the IPD problem, we would choose TD method to continue our research because of the constraint in ADP and MC that mentioned before. The research will focus on how to improve the cooperation rate between learning agents while still ca n be well adaptive to Fixed Strategy Agent. Moreover, to test the robust of the RL algorithm, the environment would become non-determined by adding noise on it. For example, agent choose defect but actually cooperate occurs.

In the long run, it is desirable to apply RL methods into the research of the navigation of multiple autonomous vehicles. The initial idea of this research is that autonomous vehicles are viewed as a multiagent system and roads are seen as scarce resource, auction mechanism might be used as a way to allocate such scarce resource. We hope that autonomous vehicles can use RL methods to learn how to bid the resource in such situation.

REFERENCES

[1] Robert Axelrod and William D. Hamilton. The evolution of cooperation, 1981.

[2] Monica Babes, Enrique Munoz de Cote, and Michael L. Littman. Social reward shaping in the prisoner's dilemma. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '08, pages 1389–1392, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[3] Koichi Moriyama. Utility based q-learning to facilitate cooperation in prisoner's dilemma games. *Web Intelli. and Agent Sys.*, 7(3):233–242, August 2009.

[4] Koichi Moriyama, Satoshi Kurihara, and Masayuki Numao. Evolving subjective utilities: Prisoner's dilemma game examples. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, pages 233–240, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[5] Koichi Moriyama, Satoshi Kurihara, and Masayuki Numao. Cooperation-eliciting prisoner's dilemma payoffs for reinforcement learning agents. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1619–1620, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.

[6] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287. Morgan Kaufmann, 1999.

[7] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[8] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[9] Tuomas W. Sandholm and Robert H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995.

[10] Ryosuke Shibusawa, Tomoaki Otsuka, and Toshiharu Sugawara. Emergence of cooperation in complex agent networks based on expectation of cooperation: (extended abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents &#38; Multiagent Systems*, AAMAS '16, pages 1333–1334, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[11] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.