Project 3 - CS7642 | Correlated-Q

Andres Miguel Menendez, amenendez6, 902969924

amenendez6@gatech.edu

Commit Hash: 674180d2ad34cb9532216154f0f7897b97a34977

INTRODUCTION

It's a saturday morning and the dew on the windows drips as my espresso presses fresh aromas across my apartment. My cat stretches her from tail to claw as I sit down to immerse myself in the dark notes of my coffee. It is at this time when I consider the day and the variety of things I could do and should do. Defined by my own values, I choose as wisely as I can. Each idea, rich with subconscious calculations, approximates my understanding of my actions to the objective value to those actions. It is here where the game of life evolves as it flourishes from our futile attempts to bootstrap our limited experiences to reach even the remoteness regions of truth. Reinforcement learning has brought forth an objective wave of understanding to this very human process of identifying the best possible action within the world of *Game Theory*.

ENVIRONMENT

Although these implications could breach even existentialism, we must first understand it in its simplest form. In the following discussion I will attempt to replicate the work of Amy Greenwald and Keith Hall in their paper *Multi-agent Correlated Q-Learning*. A *2-player* game of soccer is realized upon an environment (*figure 0)* where two players can take a series of 5 actions (N, S, E, W, sticking) in a random order. Upon collision, a player may have the opportunity to steal the ball if it is the first to move. Upon scoring a goal, A reward of +100 is given to the player that scores, and a -100 to the player which is scored upon; a *zero-sum* game. With this environment I will explore the field of utilitarian Correlated Equilibrium, Foe-Q, Friend-Q and Multiagent-Q Exploration. In my experiments, I used a modified version of Pedro Velez' ML-Soccer [1].

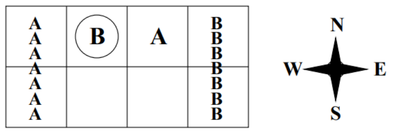


Figure 0 *- Soccer Environment*

ALGORITHMS

The several learning strategies that will be explored are methods that pursue convergence on optimal policies with respect to different attitudes. The (1) utilitarian correlated equilibrium (uCE-Q) aims to maximize the total rewards while (2) Foe-Q concerns itself on simply maximizing the selfish rewards of a player. (3) Friend-Q, antithetical to Foe-Q, looks to maximize the rewards of one player as one player consistently sacrifices itself for the good of the other. The Multagenti-Q learning algorithm is distinct as it learns on policy with a parameter that drives exploration where it chooses a deterministic action that maximizes its predicted expected reward represented in the Q-tables.

$$\text{CE-Q}(s, \overline{A}, \overline{Q}) = \max_{\pi_s \in \Delta(A(s))} \sum_{j \in N} \sum_{a \in A(s)} \pi_s(a) Q_j(s,a) \tag{1}$$

$$\text{Foe-Q}(s, Q_1, Q_2) = \max_{\pi \in \Pi(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} \pi(a_1) Q_1[s, a_1, a_2] \tag{2}$$

$$\text{Friend-Q}(s, Q_1, Q_2) = \max_{a_1 \in A_1, a_2 \in A_2} Q_1[s, a_1, a_2] \tag{3}$$

$$Q_i(s, \vec{a}) = (1-\alpha) Q_i(s, \vec{a}) + \alpha[(1-\gamma)R_i + \gamma V_i(s')] \tag{4}$$

Table 1 *- Equations*

All algorithms employed a discount factor of 0.9 and an initial learning rate and epsilon of 0.1 and 1, respectively. Decay rates were set such to drop the learning rate to 0.01 for all simulations except q-learning where the learning rate ended at 0.001 (*figure 1.1*). The epsilon strategy was similar where the resulting value would drop to .001 for all experiments (*figure 1.2*). During the simulations, the q-table was only updated every 10 steps and all Q-value differences in the figures were calculated over a step-difference of 1000 time-steps to demonstrate a clearer figure and appear more like those from the original paper. Additionally, in the case that an agent might get caught in a loop, the games were automatically reset after 50 steps.

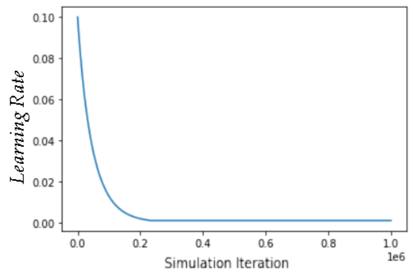


Figure 1.1 *- Learning Rate Decay*

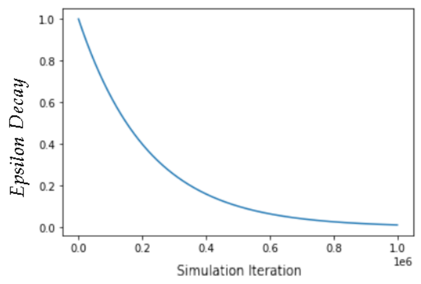


Figure 1.2 *- Epsilon Exploration Decay*

EXPERIMENTS AND RESULTS

uCE-Q

The Correlated-Q algorithm stands notable as it begs for empathy when multiple Nash equilibria drive unfairness when one player finds himself in the lesser of the equilibria. A middle man makes all the difference when approaching games as a referee can aim to improve the expected rewards of both players. By providing a signal to both players and by knowing the opponent's evaluations of states and actions, both players can maximize their expectations. This is commonly demonstrated in the game of chicken and realized in the everyday use of traffic lights. Through the use of mixed strategies both players can increase their expected rewards when provided with a recommendation. Unfortunately, because this game is zero-sum, there is never a situation where both players can increase the utilitarian value of the game.

Foe-Q

The aggressiveness of the various strategies is Foe-Q where an agent aims to selfishly improve its value by using a minimax evaluation in order to guarantee the maximum reward given that the opponent is attempting to minimize it. The results in *Figure 3* show a similar pattern to that of uCE and is due to the environment property of being a zero-sum game (this is discussed later in the DISCUSSION section). Both Q-value differences were recorded at the initial state when player A takes action S and player B sticks.

Figure 2

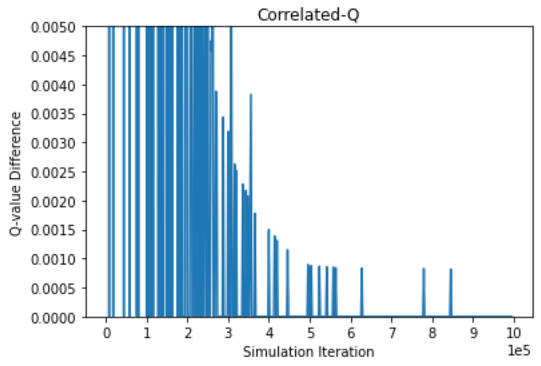
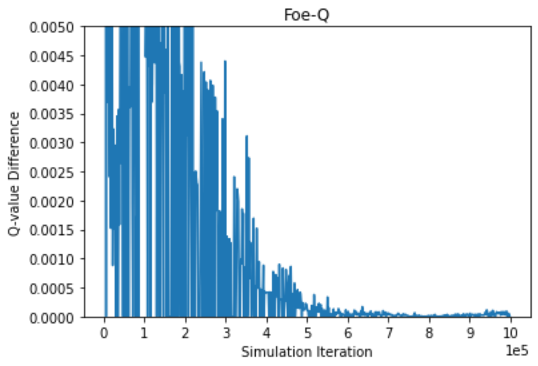


Figure 3



Friend-Q

The Friend-Q looks to maximize the reward of one player, thus, both players will effectively cooperate. This converges far faster than any other algorithm as the agents don't need to outplay each other and adapt to each other's evolving mixed strategies. Shown in *Figure 4*, the agent finds the optimal policy in about a tenth of the full duration of the learning space. The Q-value differences for friend-Q are recorded for the initial state when player A takes action S and player B sticks.

Q-Learning

Shown in F*igure 5,* the Q-learner Agent works on-policy to converge onto a deterministic optimal policy, where it acts independently of the opponents Q-table and therefore only works off a single table. The learning process resembles the value-iteration algorithm as the agent looks to take the action that is estimated to have the highest value. With a decaying epsilon driven exploration strategy, and a decaying learning rate, the agent still attempts to explore the entire environment and learn the details of the optimal policy. Unfortunately, although the q-value difference drops to near 0, it does not successfully converge. The Q-value differences for friend-Q are recorded for the initial state when player A takes action S.

Figure 4

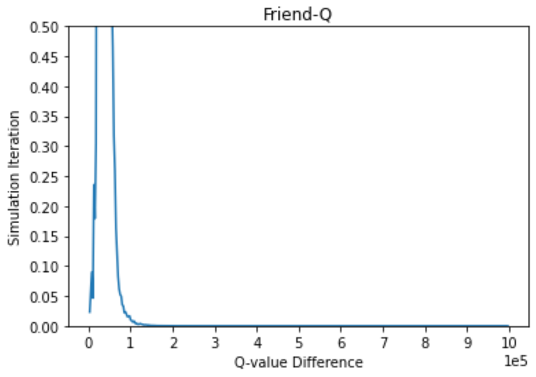
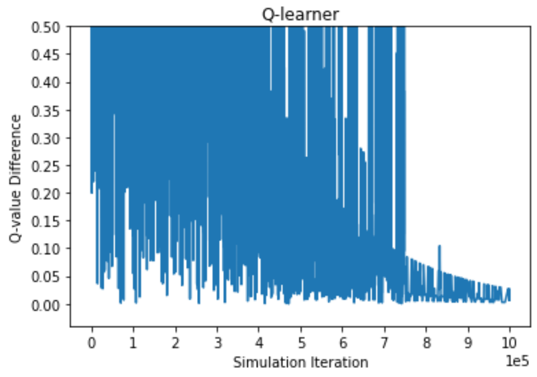


Figure 5



DISCUSSION

There are several noteworthy differences between my work and that of Greenwald and Hall's. In this discussion I will elaborate on worthwhile phenomena and reasons why my figures may not be identical to that from the original paper.

With respect to CE-Q and Foe-Q, both simulations shared similar patterns in their development. This is because the environment is a zero-sum game. The maximum cooperative reward is equal to that of the maximum competitive reward; any gain for one player is a loss of equal magnitude for the opposing player, and therefore, the maximum utilitarian value is always 0. This is consistent with the findings in Greenwald and Hall where it could be assumed that, like myself, they used the same seed for random number generation when running the two simulations.

What is most remarkable is that these two algorithms can be solved in polynomial time. Each state in the environment can be shaped into multi variable linear functions that represent mixed strategies and are driven by their objective functions that resemble those in *Table 1*. uCE-Q looks to maximize all utility while favoring the agent itself while Foe-Q takes a minimax approach of selecting the best action given a bad situation. These complex calculations can be efficiently computed through linear programming (LP) by using the Q-tables and maximizing (or minimizing the negative) with respect to an objective function. Furthermore, the resulting value of the linear program results in the optimal mixed-strategy for that time-step and an estimation of the state's value (the expected reward given that mixed-strategy) that is bootstrapped into the Q-table update. These calculations can be powerful but tricky to formulate as various constraints play important roles.

Figure 3, Foe-Q, resembles the pattern from the original paper, but, *Figure 2,* appears to take on a limited more discrete development. This is likely due to the linear programming setup where the solver would return "None" values when no solutions were found for that LP system. These exceptions are handled by evoking a random action on behalf of both players and avoiding a Q-update for that time-step. The agent already only updates the Q-table every 10 episodes and altogether the CE-Q agent ends up returning a somewhat empty history of Q-updates. The convergence is accurate as both Q-tables in CE-Q and Foe-Q, stop changing significantly, independent of the decayed learning values. Greenwald and Hall's simulations must have used a more robust LP system where the agents would handle exceptions in different ways in order to create a more continuous learning experience for the agents.

Figures 2 and 3 were also notably different to the figures in the original publication when considering their y-axis ranges. Figures 2 and 3 were bound to a max value of 0.005 on the y-axis while the original document worked with figures that scaled to a Q-value difference between 0 and 0.5. Although the maximum values peaked way over 0.005, the insightful data and subtle variations occurred in this region. By noting the similar pattern between the Foe-Q and the original figure, it is clear that the simulations and data are behaving similarly and as they are meant to. Yet, the actual resulting data is scaled down by 100. This may be due to subtle differences between the Q-table update equations and could be easily "fixed" by multiplying the entire data-set by 100. It is not clear which exact parameter is compelling this phenomenon but it may be dismissed as the actual behavior of the agents is what matters and is correct as they resultantly converge on their optimal policies.

Friend-Q was solved by simply treating the q-tables as a cooperative set of actions where the agent chooses an action for both players in order to maximize the reward for one player. This result aligns very closely to the results of Greenwald and Hall as the convergence occurs around 100,000 steps. My agent appears to take a moment longer to see significant changes in its q-values as it ramps up slower than the original figure. This may simply be due to the randomness of using a mixed strategy.

In the final figure, the agent's Q-value differences begin to approximate 0 at the end but are solely due to the decaying learning rate. If it would reach true convergence then the actual Q-values wouldn't change very much independently to the learning rate. This agent would have widely ranging Q-value differences but do not fluctuate the q-tables because of the small learning rate. Overall, *Figure 5* shows a very close resemblance to the original figure in Greenwald and Hall's publication. One big immediately noticeable difference is the learning-rate decay schedule. The original agent appears to delay the decay of the learning rate but then aggressively decays towards the end as the sloping tail appears to be greater than than this agent's decay rate. Both still demonstrate a turbulent learning pattern where neither appear to reach consistent learning independently of the learning rate.

CONCLUSION

Replicating these figures was not only non-trivial but intimately difficult and frustrating. Each agent is bound to and incredibly sensitive to every possible hyperparameter. When considering the combinations between all the hyperparameters and the various ways of plotting the data, it can be immediately apparent how many different ways there are to show the findings of Greenwald and Hall. Although they clarified several of their hyperparameters, it was very difficult to know exactly how they implemented their update methods. At what frequency was the agent updating its tables, or what exact Q-value difference were they plotting on their figures? Implementing Linear Programming and the algorithm themselves, was already difficult. But adding the intermittent updates and the expanded Q-value differences.. it was soul-ripping and near-impossible to trust that my agent was doing what it was supposed to be doing. It is also worth piling in the frustrating nature of a 3 hour long runtime for CEQ to truly express the torment of this project.

Through meticulous care it was possible to converge on the right combination of hyperparameters and I am content with the level of fidelity I achieved as I was able to run each experiment several times in order to demonstrate consistent behaviors and choose the seed that resembled the original figures most closely. Nevertheless, putting aside the lack of clarity, this paper successfully demonstrated the deeper implications of these algorithms as they could be incredibly powerful and useful in their right applications even if these discoveries may only be limited to the paradigms of 2 player, constant-sum games.

REFERENCES

[0] Amy Greenwald, Keith Hall. 2003: Correlated-Q Learning, Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[1] Velez, Pedro. ML_Soccer. https://github.com/pdvelez/ml_soccer

[2] Amy Greenwald, Keith Hall, and Martin Zinkevich. 2007. Correlated Q-Learning, Journal of Machine Learning Research 1 (2007) 1-1.

[3] Michael L. Littman. Friend-or-Foe Q-learning in General-Sum Games, AT&T Labs Research, 180 Park Avenue, Florham Park, NJ 07932-097

[4] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning, Brown University / Bellcore Department of Computer Science Brown University Providence, RI 02912-1910