Soccer Project

Chenxi Yu

Sunday, July 23, 2017

1. Introduction

This project presents a replication of both Greenwald and Littman's multi-agent Q learning algorithms: Correlated-Q, Foe-Q and friend-Q. The paper has inspired me to use Q-tables and apply linear programing to make agents optimal policies converge to game theoretic equilibria. 4 figures from the Greenwald's Correlated-Q learning will be replicated below. Further analysis and conclusions based on these results will be examined.

The algorithm indicated in Table 1 in the paper demonstrates how each Q learner works in the 2 by 4 soccer's game environment. In this project, I follow its setting, allowing the agent act randomly among 5 actions: N, S, E, W and X(stick). The paper also specifies a game rule: "If this sequence of actions causes the players to collide, then only the first moves. But if the player with the ball moves second, then the ball changes possession "(Greenwald). The zero-sum game characteristics is reflected by the gain or loss of 100 on an agent based on a game result. Most importantly, this game exhibits no deterministic equilibrium policies since each agent could suffer infinite blocking by the other. Thus, in my implementation, I have kept a Q-table by 2*8*5*5 for the Q-algorithms introduced in this paper with separated 2*8*5 for the basic Q learner.

MULTIQ(MarkovGame, f, γ, α, S, T)

Inputs	selection function f
	discount factor γ
	learning rate α
	decay schedule S
	total training time T
Output	state-value functions V_i^*
	action-value functions Q_i^*
Initialize	$s, a_1, \ldots, a_n$ and $Q_1, \ldots, Q_n$

for $t = 1$ to T

1. simulate actions $a_1, \ldots, a_n$ in state s
2. observe rewards $R_1, \ldots, R_n$ and next state s'
3. for $i = 1$ to n
   (a) $V_i(s') = f_i(Q_1(s'), \ldots, Q_n(s'))$
   (b) $Q_i(s, \vec{a}) = (1-\alpha)Q_i(s, \vec{a}) + \alpha[(1-\gamma)R_i + \gamma V_i(s')]$
4. agents choose actions $a_1', \ldots, a_n'$
5. $s = s'$, $a_1 = a_1'$, $\ldots$, $a_n = a_n'$
6. decay α according to S

$$\text{Nash}_1(s, Q_1, Q_2) = \max_{a_1 \in A_1, a_2 \in A_2} Q_1[s, a_1, a_2] \quad \text{Nash}_1(s, Q_1, Q_2) = \max_{\pi \in \Pi(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} \pi(a_1) Q_1[s, a_1, a_2]$$

On the left is the friend's Q , on the right the foe's Q and the bottom right is the CE-u Q

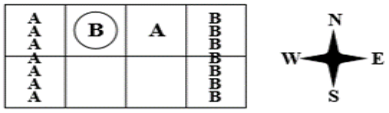


Figure 4. Soccer Game. State s.

maximize the *sum* of the players' rewards:

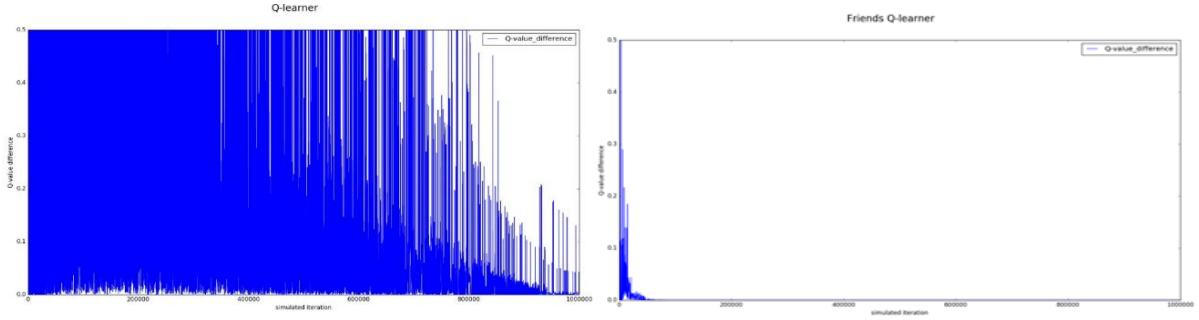
$$\sigma \in \arg\max_{\sigma \in \text{CE}} \sum_{i \in I} \sum_{\vec{a} \in A} \sigma(\vec{a}) Q_i(s, \vec{a})$$

As studied in multiple sources, Q learning has remarkable properties of global convergence, yet indicated by Greenwald that Q learners in general sum game may depend on initial conditions. This is especially apparent if the agent is in a non-deterministic environment or state existing no deterministic equilibria. This leads to a reasonable performance benchmark: the error term reflecting corresponding to State s with player A taking action S and player B sticking.

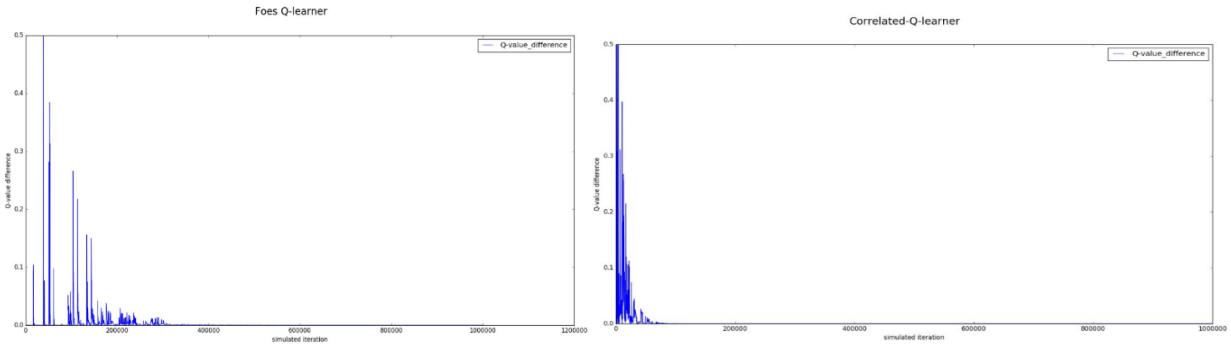
However, there are a couple of uncertain factors in replicating the results. The paper in page4 mentions “Our experiments reveal that off-policy correlated-Q, foe-Q, friend-Q ($\alpha \rightarrow 0.001$ and $\gamma = 0.9$.) and on-policy Q-learning (i.e., epsilon-greedy, with epsilon $\rightarrow 0.001$, $\alpha \rightarrow 0.001$, and $\gamma = 0.9$) all converge empirically in the three grid games”(Greenwald) Therefore, my parameters refer to this general description since there are no other spot covering better detail. Regardless, the decay methods of epsilons and alpha are a myth so far. Thus I just use regular exponential decaying trick learned in class by 100000 times roughly according to my q –learners 1000000 iterations. Thus, my decaying factor is .99992 since I roughly observed the iterations (number of steps) is around 90000 in updates. Thus, it make sense to have a very close graph as below in regular q-learners. This also raises another unknown—the meaning of iterations. Yet, according to my trials and errors, it make more sense to assume iterations as number of steps of each players instead of the number of updates. Otherwise, the graph would not likely differentiate the convergence difference of each q–learner, since Correlated-Q, Friend’s Q and Foe’s Q are all converging fast unlike Q-learner as demonstrated below. Moreover, how to random start the game is also unknown. Yet, due to the purpose of this experiment, I assume the starting point is State s.

Another challenge I have for implementing the project is the matrix setup for the cvxopt solvers operations. I have referred to how things work in the chicken dare and rock paper scissors game and end up implementing the minimax algorithm with (12,6),(12,1) and (6,1) matrix and the CE-u with (67,25),(67,1) and (25,1) matrix on glpk.

2. Analysis and Discussion



The figure exhibited on the left above is Q-learning’s performance, as expected, it has not converged after 1m iterations. Meanwhile, the Friend’s Q on the right is converging quickly exactly as indicated in the paper. The Q-learning like many other experiments I have done show that it is not converging, but because its learning rate is decreasing like the hidden curve appears within the splines. This is consistent with Greenwald’s figure. Friend’s Q also converges to a deterministic policy for Play B at State s, yet this behavior does not seem realistic or rational. In terms of the speed, they are both faster than Foe’s and CE-u since it is simpler to implement by mostly querying to get value updates.



On the Left above is the Foe's Q, while on the right is the CE-u Q. They are both converging fast while CE-u converges just as fast as the friend's Q, though the 1st trial I did was not as fantastic as this appears above. Noticeably, the CE-u was running relatively slower at the beginning and speeding up fast after certain thresholds. That may be a sign of convergence. CE-u Q does take a significant longer time, yet it converges to correlated equilibria and performs more robustly than the Foe's Q and rational than friend's Q, though performance-wide their convergence look similar. I have also checked the Q –value table and found that the EC-u and Foe's are identical, this also coincides with the paper's conclusion—CE-Q learns minimax equilibrium policies in the 2 players zero sum game. It is worthwhile to learn that they could be implemented via linear programming in cvxopt. They also show optimal outcome most of the time. However, we observe a significant difference in the graph between the paper and mine. This could be explained by the convergence rate of CE-u, Friend's and Foe's somewhat depending on the stochastic process and the learning rate with the decay rate. Intuitively, if the randomized process in certain period gears toward some player and coincidentally the game episodes result closer to the equilibria, it may converges faster than the otherwise.

However, when it comes to the possible different parameters decay affecting the outcome, I do not find it make a significant difference. Since as figures shown above, they all converges apparently faster than the decay rate of the learning rate which is scheduled to die out the learning rate within 155100 updates(alpha made lower than .001), though we could not foresee the learning rate decays very close to 0.001. Thus, I believe a faster decaying would not make CE-u, Foe's and Friend's perform much worse.

3. Conclusion

Overall, my replication shows consistent results with the paper. In a deterministic game, Q learning converges better mostly because how fast it runs compared with other Q learners, However, it may get stuck given sufficient noise or failing to converge in a stochastic game setting like in this project. Foe's Q is known as minimax Q is able to converge to equilibrium policies analytically while QE-u converges to policies similar to minimax Q. It is also easier to compute unlike Nash-Q while unlike Friend's Q, it is consistent with the AI view of individually rational behaviors.

Video link: https://www.dropbox.com/s/owfl04tszhri6w1/Project3.mp4?dl=0

Greenwald Amy, and Hall Keith. "Correlated-Q learning." (2002): n. pag. Brown University. Web.

M. Littman. Friend or foe Q-learning in general-sum Markov games. In Proceedings of Eighteenth International Conference on Machine Learning, pages 322– 328, June 2001.