Project 3 Report

Yihuan Su

Git hash: b28c65b40ac9816795e508fdfd249217b52e81ba

I Introduction

As discussed in class, correlated equilibrium is a more general solution conecept than Nash quilibrium. With correlated equilibirium, players of a game would follow actions assigned by a information object. Greenwald and Hall (2003) introduced a multiagent Q-learning algorithm based on correlated equilibrium, namely Correlated-Q (CE-Q) learning, which is a generlization of Nash-Q and Friend-and-Foe-Q. Both Nash-Q and Friend-and-Foe-Q are due to earlier attempts to design algoirthms that are capable of learning equilibrium policies in multiagent general-sum Markov games. However, Nash-Q can converge to Nash equilibrium policies under some restrictive conditions; Friend-and-Foe-Q only learns equilibrium policies in certain classes of games. The focus of this report is to discuss my attempt to replicate the soccer game experiment in paper of Greenwald and Hall (2003). Before going into details of the experiment, I first lay the ground by introducing some key concepts of Markov games, correlated equilibrium and multiagent Q-learning.

II Markov Games

A Markov game is a stochastics game for which the probability transitions satfisfy the Markov property. The Markov decision process that we've been discussing in the class is essentially a one-player Markov game. Recall Bellman's equation that represents the optimal state-action values for an MDP

$$Q^*(s,a) = (1-\gamma)R(s,a) + \gamma \sum_{s'} P[s'|s,a]V^*(s') \qquad (1)$$

where $0 \leq \gamma < 0$ is the discount rate, *R(s,a)* is the reward at state *s* given action *a*, *P[s'|s,a]* is the transtion probability to state *s'* given state *s* and action *a*, and *V*(s)* is the state value fucntion

$$V^*(s) = \max_{a \in A(s)} Q^*(s,a) \qquad (2)$$

where *A(s)* is the set of action at state *s*. Based on the state value function, the potimal policy can be defined as

$$\pi^*(s) \in \arg\max_{a \in A(s)} Q^*(s,a) \qquad (3)$$

In multiagent Markov games, palyer *i*'s state action value function is defined as

$$Q_i(s,\vec{a}) = (1-\gamma)R_i(s,\vec{a}) + \gamma \sum_{s'} P[s'|s,\vec{a}]V_i(s') \qquad (4)$$

where $\vec{a} = (a_1, ..., a_n)$ is a vector of actions taken by each player. Based on intuition, unlike MDP, the concept of maximizing a player's reward with respect to actions is not adequate for multiagent Markov games because the actions of players may not be deterministic. For example, in the game rock-paper-scissors, players take actions simultaneously. A player could take actions randomly (Greenwald and Hall 2003).

III Correlated Equilibrium

A Nash equilibrium is a set of independent strategies or policies in which all players' rewards are optimized with respect to one another's strategy. It can be represented as as a vector of independent probability distributions over actions. A correlated equilibrium permits dependencies among players' probability distribution while ensuring all players' rewards are optimized. With a correlated equilibrium, players follow a strategy based on a probability distribution over the combinations of actions. For example, a traffic light is a correlated equilibrium. For two cars/players at an intersection, a traffic light will either signal actions (GO, STOP), or (STOP, GO). It's optimal for both players to follow the signals from a traffic light, or the players could crash into each other or stuck at the intersection (Greenwald and Hall 2003).

As discuss in the class, Nash equilibria are often hard to find, while correlated equilibria can be computed in polynomial time via linear programming.

IV Multiagent Q-Learning

Based on Nash equilibrium, Hu and Wellman (1988) proposed Nash-Q for the general case of multiagent, general-sum games of which state value function defined as

$$V_i(s) \in NASH_i(Q_1(s), ..., Q_n(s)) \qquad (5)$$

where $NASH_i(Q_1(s),\ldots,Q_n(s))$ is the ith player's value function according to some Nash equilibrium with respect to the Q functions at state *s*.

Motivated by Nash-Q, Littman (2001) introduced Friend-and-Foe-Q. For simplicity, the two-player case of Friend-and-Foe-Q is defined by

$$NASH_1(Q_1(s), Q_2(s)) = \max_{a_1 \in A_1, a_2 \in A_2} Q_1(s, a_1, a_2) \qquad (8)$$

if the opponent is a friend and

$$NASH_1(Q_1(s), Q_n(s)) = \max_{\pi \in \Pi(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} \pi(a_1) Q_1(s, a_1, a_2) \quad (9)$$

if the opponent is consider as a foe. One can see that equation (8) is maximizing any player's reward. It learns what's called coordination equilibrium. Equation (9) aim to maximize the reward of player 1 with the assumption that player 2 will take actions to minimize the reward of player 1. It learns a adversarial equilibrium.

Greenwald and Hall (2003) proposed correlated-Q learning based on correlated equilibrium. The state value function is defined as:

$$V_i(s) \in CE_i(Q_1(s), ..., Q_n(s)) \quad (10)$$

where $CE_i(Q_1(s),\ldots,Q_n(s))$ is the ith player's value function according to some correlated equilibrium with respect to the Q functions at state *s*. Given

$$CE_i(Q_1(s), \ldots, Q_n(s)) = \sum_{\vec{a} \in A} \sigma(\vec{a}) Q_i(s, \vec{a}) \; where \; \sigma \in CE \; is \; the \; policy$$

Greenwald and Hall (2003) introduced four variants of correlated-Q learning according to different mechanisms of correlated equilibrium selection:

1. utilitarian (uCE-Q), maximize the sum of the players' rewards

$$\sigma \in \arg\max_{\sigma \in CE} \sum_{i \in I} \sum_{\vec{a} \in A} (\vec{a}) Q_i(s, \vec{a}) \quad (11)$$

2. egalitarian (eCE-Q), maximize the minimum of the players' rewards

$$\sigma \in \arg\max_{\sigma \in CE} \min_{i \in I} \sum_{\vec{a} \in A} (\vec{a}) Q_i(s, \vec{a}) \quad (12)$$

3. republican (rCE-Q), maximize the maximum of the players' rewards

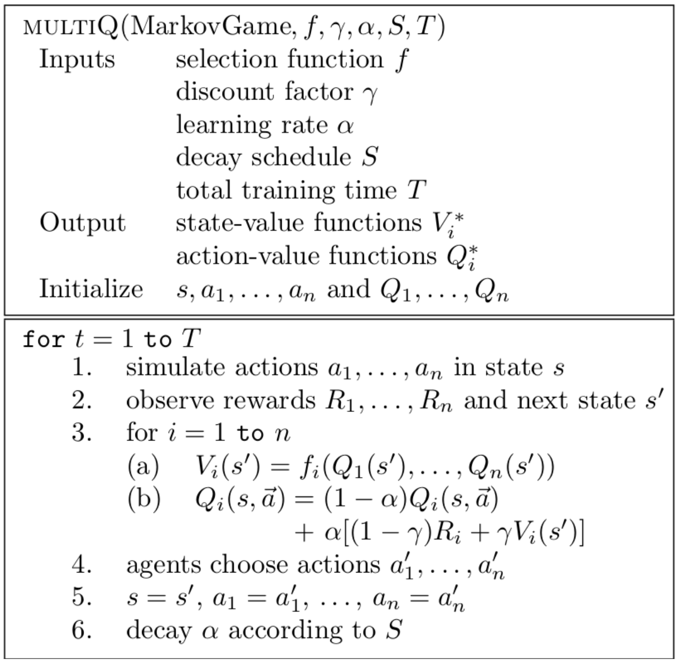
$$\sigma \in \arg\max_{\sigma \in CE} \min_{i \in I} \sum_{\vec{a} \in A} (\vec{a}) Q_i(s, \vec{a}) \quad (13)$$

4. libertarian (lCE-Q), maximize the maximum of each individual player i's rewards

$$\sigma^i \in \arg\max_{\sigma \in CE} \sum_{\vec{a} \in A} (\vec{a}) Q_i(s, \vec{a}) \; where \; \sigma = \prod_i \sigma^i \quad (14)$$

All four objective functions can be solved using linear programming. Note in order to implement these objective functions., it's necessary for Q-tables of all players to be transparent to one another. Therefore, the method is only applicable to games in which each player's actions and rewards are observable by all the players.

Table 1 below is taken from Greenwald and Hall's (2003) paper. It describe a generic multiagent Q-learning algorithm. For each of the approach described above, we just need to change the formulation of value function V_i accordingly.



```
MULTIQ(MarkovGame, $f$, $\gamma$, $\alpha$, $S$, $T$)
  Inputs      selection function $f$
              discount factor $\gamma$
              learning rate $\alpha$
              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. $s = s'$, $a_1 = a'_1$, ..., $a_n = a'_n$
    6. decay $\alpha$ according to $S$
```

Table 1

V Soccer Game

The soccer game was first introduced by Littman (1994). It a zero-sum grid game for which no deterministic equilibrium policy exist because there is no deterministic equilibrium at certain states. For the game described in Greenwald and Hall (2003), the soccer field is a grid as shown in Figure 1. There are two players A and B. The five possible actions for each players are: N(go up), S(go down), W(go left), E(go right), and stick(do nothing). Both players pick their actions simultaneously, but they take actions in random order. A player can not move into a cell that's occupied by the other player. The ball (indicated by the circle in Figure 1) would change possession if and only if the player with the ball attempt to move into the other player. If the player with the ball move into his own goal (indicated by 'AAAAAAA' and 'BBBBBBB' in Figure 1), he gets +100 points and his opponent gets -100 points. If the player with the ball move into his opponent's goal, he gets -100 points and his opponent gets +100 points. The game ends in either case.

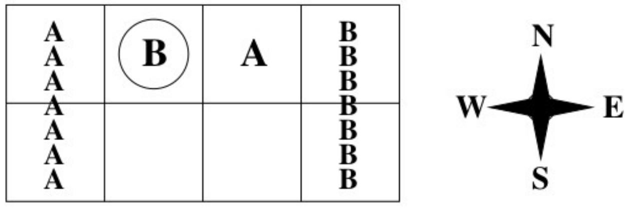


Figure 1

Greenwald and Hall (2003) didn't describe the consequence of an illegal move. For example, what will happen if a player tries to take an action to hit the edge of the field. A reasonable assumption is that the player stays where he is when making an illegal move.

Greenwald and Hall (2003) also didn't mention explicitly what's the initial state of the game. Because of how the error term is caculated, which will be discussed in the following section, it makes intuitive sense to consider state depicted in Figure 1 (state s) as the initial state.

As mentioned earlier, in this game, deterministic equilibria don't exist at some states. The states illustrated in Figure 1 is one of those states. If the policy is deterministic, player B will just get blocked by player A forever. If player B follows a nondeterministic policy, then there will be chance that player B can pass player A.

VI Experiments and Restuls

VI.A Expperiments and Assumptions

The goal of this project is to replicate the graphs in Figure 3 in Greenwald and Hall (2003). Greenwald and Hall (2003) experimented utilitarian correlated-Q learning, Foe-Q, Friend-Q, and on policy Q-learing in the soccer game. I impletemented the algorithms in Python based on the description in Table 1 and conduct the experiment described in Greenwald and Hall (2003). As we are focusing on player A's Q-values, in my impletementation, player B always takes random action. Figure 2, 3, 4, 5 are my attempt to replicate the graphs in Greenwald and Hall (2003).

The values of x-axis of the graphs represent iterations or time. Greenwald and Hall (2003) didn't say explicitly if each iteration represents one episode of the game or one step of the game. However, based on the description in the paper, my understanding is that one iteration represents one time step of the game. In each experiment, I run 1 million simulation iterations like what's shown in Greenwald and Hall (2003). In my implementation, if the game ends, it restarts from the initial states, and the execution won't finish until it reaches 1 million time steps.

The values of y-axis represent the error terms $ERR_i^t = |Q_i^t(s, \vec{a} - Q_i^{t-1}(s, \vec{a})|$. In Figure 2, 3, and 4, $Q_i^t(s, \vec{a})$ corresponding to player A's Q-value at state s (shown in Figure 1), with player A taking action S and player B do nothing. In Figure 5, $Q_i^t(s, \vec{a})$ corresponding to player A's Q-value at state s (shown in Figure 1), with player A taking action S. Note given how the algorithms are impletement, the specific Q-value don't get udpated unless the game is at state s and players take the actions mention earlier, or else the error term will be 0. Therefore to replicate the graphs in Greenwald and Hall (2003), it makes intuitive sense to consider state s as the initial state so the specific Q-value gets updated more frequently.

Greenwald and Hall (2003) also failed to specify all the hyper-parameter values used in the experiments. Given Greenwald and Hall (2003) mentioned they experimented the same set of algorithms used in the grid games, it's safe to assume the hyper-parameter values are the same as well. Therefore, I have discount rate γ=0.9, ϵ converge to 0.001, α converge to 0.001. For the rest of the hyper-parameter values, based on some experimentation and intuition, I adopted initial α=1.0, α decay=0.999994, initial ϵ=0.9, ϵ decay=0.999994 for all four algorithms.

VI.B Results

Compared to graphs in Greenwald and Hall (2003), my replicated plots (Figure 2, 3, 4, 5) are somewhat different. Greenwald and Hall (2003) lacks details of how are the graphs created, so it's hard to recreate the exact same graphs. However, the general patterns of my replicated graphs resembles the plots in the original paper.

Similar to the graphs in Greenwald and Hall (2003), the error term plots of Correlated-Q and Foe-Q are almost idenitical, As shown in Figure 2 and Figure 3. Both alogrithms eventually converge at about 800,000 iterations. That's a bit different from what's shown in Greenwald and Hall (2003). Most likely it's due to different hyper-parameter values. Nevertheless, they tell the same sorry. In the soccer game, player A and player B are competing opponents. Foe-Q's minimax approach is well suited for this game. However, Correlated-Q is able to learn the same non-deterministic policy.

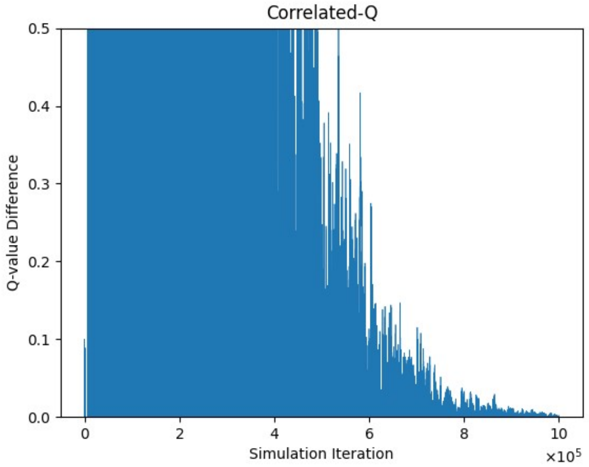


Figure 2

Friend-Q (Figure 4) behaves in the same way shown in Greenwald and Hall (2003). It converges almost immediately to a irrational deterministic policy. As Friend-Q assumes both players are working together to maximize each other rewards. At state s, player A assumes player B will take action W and score for him. While player B will take action E and pass the ball to player A, assuming player A will score for player B.

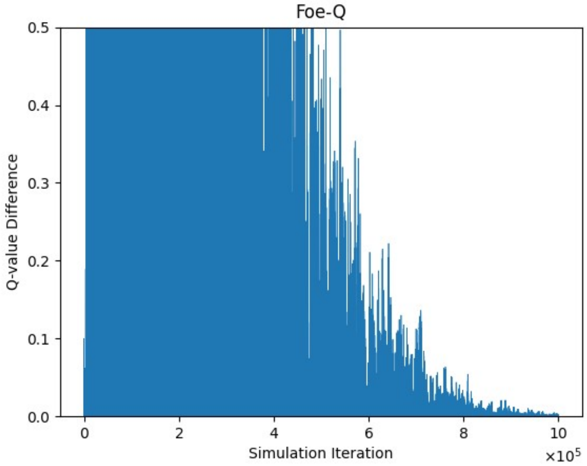


Figure 3

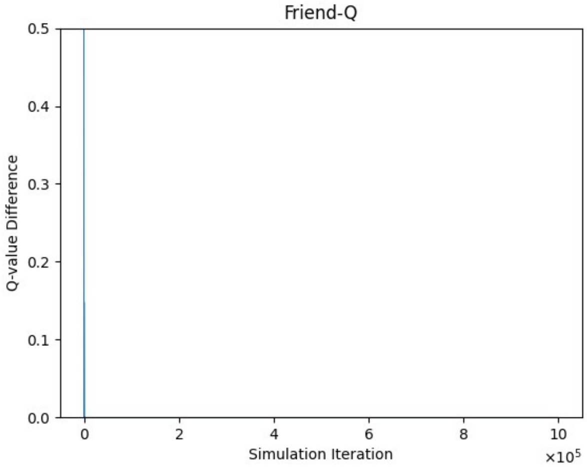


Figure 4

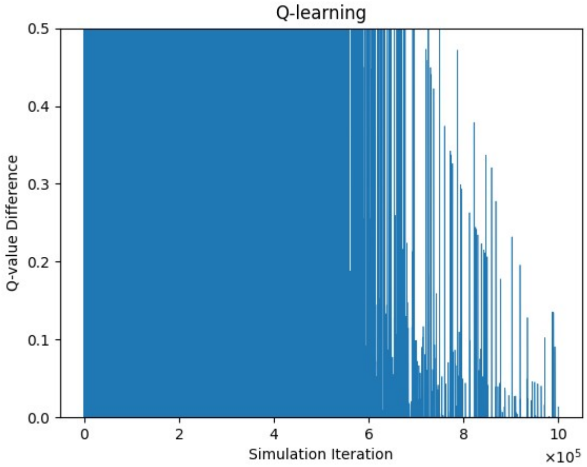


Figure 5

As shown in Figure 5, in the soccer game, Q-learning does not converge, which is in accordance graph shown in Greenwald and Hall (2003). Even though the Q-value difference decreases as the number of interations increases, that’s simply because the learning rate is decaying over iterations. Q-learning algorithm only consider one player’s rewards and actions. In a multiagent game, other players actions is a crucial factor to consider. At a given state, the same action taken by one player could result in very different rewards when other players take different actions. Greenwald and Hall (2003) pointed out that Q-learning seeks to converge to optimal determisnistic polices, however, as mentioned earlier, in the soccer game at state s, no such policy exist.

VII Conclusion

The objective of this project is to replicate the soccer game experiment and related graphs in Greenwald and Hall (2003). Based on the paper, in this report, I briefly introduced Markov games, Correlated equilibrium and Nash equilibrium. Then I walked through some development in multiagent Q-learning as well as Correlated-Q which is proposed in Greenwald and Hall (2003). Later, I disccussed the soccer game environment and the experiments conducted, as well as the results of the experiments. My findings are in accordance with Greenwald and Hall (2003). They show in a zero sum multiagent non-cooperative game such as the soccer game in which no deterministic equilibrium policy exist, correlated-Q algorithm can converge to the same policy as Foe-Q. Given it’s assumptions, Friend-Q converges to an irrational policy. While Q-learning is guaranteed to converge to deterministic policies in MDPs under the assumption that all state-action paris are updated continuously, it failed to converge in the soccer game because no optimal deterministic policy exist. Correlated-Q is more general than Q-learning and Friend-and-Foe-Q. For a competitive game like soccer, Friend-Q is ill-suited. Foe-Q conveges to a non-deterministic policy, and Correlated-Q converges to the same policy. Greenwald and Hall (2003) also showed in other types of games where Friend-Q or Foe-Q performs poorly and Q-learning performs well, Correlated-Q can still performs really well.

VIII Refernces

1 Greenwald, A., & Hall, K. (2003). *Correlated-Q learning.* In Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03). AAAI Press, 242–249.

2 Littman, M. L. 2001. *Friend-or-Foe Q-learning in General-Sum Games*. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 322–328.

3 Littman, M. L. 1994. *Markov games as a framework for multi-agent reinforcement learning*. In Proceedings of the Eleventh International Conference on International Conference on Machine Learning (ICML'94). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 157–163.