RoboCup Correlated-Q Edition:

Shailesh Tappe: stappe3 **Git Hash: 785637d865a1cf9c5ac2a49f2bbaa2f8ebd1721b**

Objective: The objective of this project report is to explore and replicate multiagent Q-learning algorithm based on correlated equilibrium (CE) solution concept, put forward by Amy Greenwald and Keith Hall in their "Correlated-Q Learning" paper published in 2003. The focus of report is to apply Correlated-Q, Friend-Q, Foe-Q and classic Q learning to the "Soccer Game" example discussed in the paper.

Introduction:

1. **Markov Games:** Markov or stochastic games are multi-player MDPs and is represented as tuple <I, S,(Ai(s)), P,(Ri)>, where I is set of number of players, S as set of states, Ai(s) as joint action vector for the players, P the probabilistic transition function and Ri(s, ~a) as the ith player's reward. Instead of one player action, actions are considered to be joint vector for all the players.
2. **Correlated Equilibrium:** The paper presented by Greenwald and Hall extends idea of multiagent learning technique Correlated-Q learning. This method generalizes Nash-Q presented by Hu and Wellman in 1998, as well as Friend-Q & Foe-Q presented by Littman in 2001. Expanding on Nash-Q algorithm, which is based on Nash Equilibrium (NE), learns via vector of independent PD (probability distribution). These independent PD vectors is not effective in computation for Nash equilibria, this is where Correlated equilibrium (CE) is helpful, as it represents PD over join space of action simply by using linear programming.
3. **Multi-agent Q-Learning:** Single agent Q-learning for MDP is generalized for multiple agents using Q updates for a particular agent I is based on equation 1 figure 1. The most critical part of update which governs type of equilibrium being applied is based on equation 2 for Foe-Q, equation 3 for Friend-Q, and equation 4 for CE-Q learning algorithms.

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

$$V_1(s) = \max_{\sigma_1 \in \Sigma_1(s)} \min_{a_2 \in A_2(s)} Q_1(s,\sigma_1,a_2) = -V_2(s) \quad \text{(Equation 2)}$$

$$V_i(s) = \max_{\sigma \in A(s)} Q_i(s,\vec{a}) \quad \text{(Equation 3)}$$

$$\sigma \in \arg\max_{\sigma \in \text{CE}} \sum_{i \in I} \sum_{\vec{a} \in A} \sigma(\vec{a}) Q_i(s,\vec{a}) \quad \text{(Equation 4)}$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha \underbrace{(R' + \gamma \max_{a} Q(s',a))}_{\text{learned value}} - \underbrace{Q(s,a)}_{\text{old value}} \quad \text{(Equation 5)}$$

$$\text{ERR}_i^t = \left| Q_i^t(s,\vec{a}) - Q_i^{t-1}(s,\vec{a}) \right| \quad \text{(Equation 6)}$$

Figure 1

Greenwald and hall utilized these equations to convey the results of multi-agent Q-learning along with regular Q-learning (equation 5) . All these equations is applied to Two-Player, Zero-Sum Markov game in the form of grid soccer and compare how these converge based on error metric as described in equation 6.

Approach: The soccer game environment was initialized as a two-player zero-sum game, with player starting positions randomized. Each player has these possible actions- North, South, East, West and stick. The players' actions are randomized. If a player with the ball moves into the goal position, he scores +100 if it is his own goal and the other player scores -100, or he scores -100 if it is the other player scores

100, or he scores -100 if it is the other player's goal and the other player scores+100. In either case, the game ends.

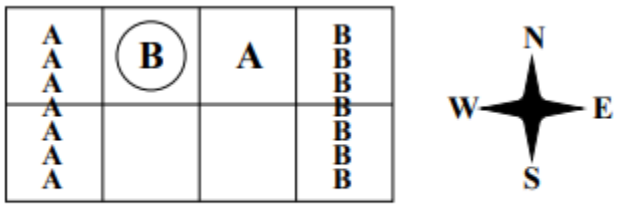
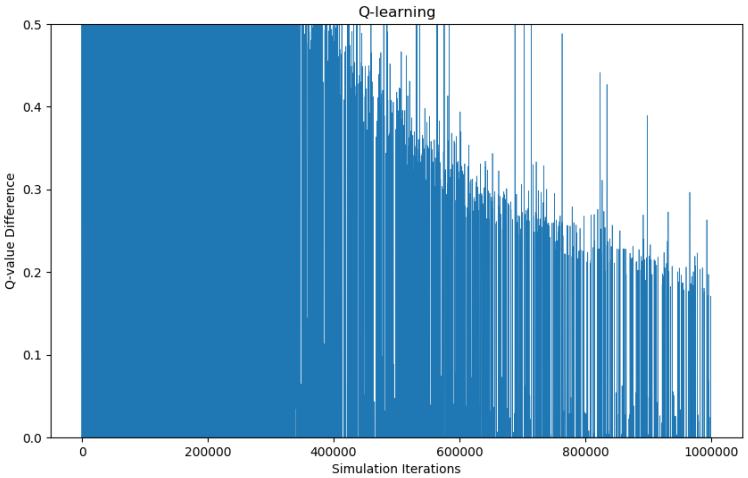


Figure 2

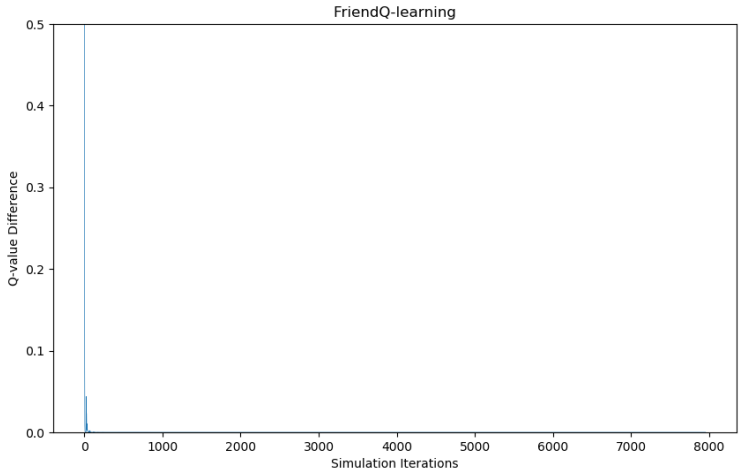
The soccer grid world is represented by 2X4 matrix and game always initiate with ball ownership with player B. Second state of the game is when B stays at initial position and A moves to south with joint position as [South, Stick]. This lead both the players chose actions and executing them randomly to follow moves that lead to player winning the game by achieving goals (opposing or own goal).

Experiments: Based on paper, four experiments were to implemented to prove that Friend-Q, Foe-Q and CE-Q algorithms converges while Q-Learning fails to converge in the multi-agent scenario even after 1000000 iterations of learning.

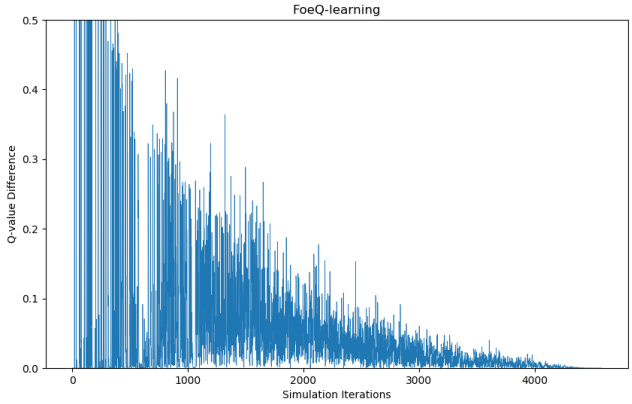
1. **Q-Learning:** While updating Q Learner, formula $Q_i(s,\vec{a}) = (1-\alpha)Q_i(s,\vec{a}) + \alpha[(1-\gamma)R_i + \gamma V_i(s')]$ is referenced, which states that Q-table must be updated from join action space. However, when Q-learner picks up best action for update, "the best" action from joint space is the one that benefit both the players, turning Q-learner to Friend-Q. This contradicts with points author were trying to make, since Q-learner is supposed to be isolated and never converge due to lack of knowledge of another agent. This approach demonstrates that traditional Q-learning can not lean in multiagent environment by not converging.



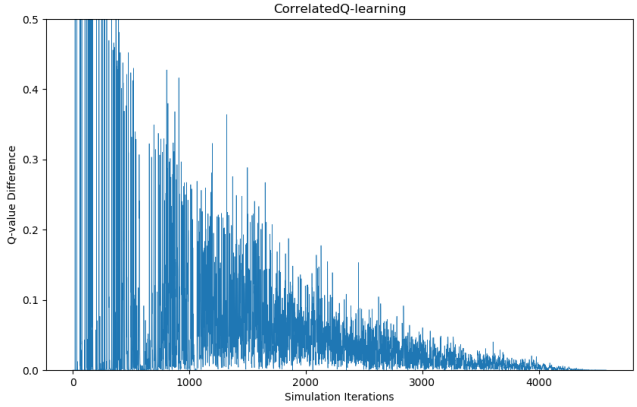
2. **Friend Q:** Friend Q was presented by Littman in 2001 paper and this learning technique works on principle of coordination equilibrium between agents in play. As actions are in form of joint pair (players take simultaneous actions to move different states), the objective is to simply maximize Q values in the given state for actions taken by both players, hence for Friend-Q algorithm no linear programming is required.
   In another word, Friend-Q naively assumes that opponent will take action that would maximize his rewards, to the point that opponent to score himself.



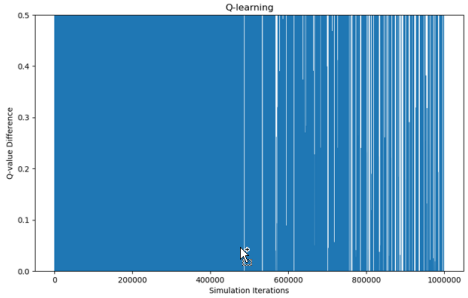
3. **Foe-Q:** Foe-Q, based on Minimax-Q algorithm presented by Littman 1994, the goal was to maximize the worst case (minimum) over all that the opponent can play. This algorithm was implemented using the information provided in Littman 1994.
   In another words, Foe-Q learner utilizes adversary strategy, in which each agent picks up action that minimizes opponent's reward, even if it is not best choice for himself. COXOPT library is used for linear programming minimizing equation. Minimizing function takes Q[s] and outputs probabilistic action space (PAS) that minimizes agent's rewards based on opponent's actions. The best action is then selected by agent with respect to PAS and Q table is updated accordingly.



4. **CE-Q**: CE-Q learner strategy uses correlated equilibria. At each state, correlated equilibria is calculated using linear programming. Utilitarian CE-Q (uCE-Q) is implemented with this experiment, in which correlated equilibria (CE) maximizes sum of both agent's rewards.



- **Issues and Resolution:** There was a stiff learning curve in understanding CE-Q and Foe-Q algorithms and also linear programming aspect of this experiment. Besides this stiff learning curve, numerous experiments were carried out to adjust hyperparameters, specifically epsilon decay and alpha decay for Foe-Q and CE-Q algorithm as it was sensitive with result outcome.
- **Observations/Comparisons:** The generated plot for Friend-Q very closely resembles with plot in paper. It can be clearly seen that algorithms converge well before 100,000 iterations coinciding with original plot.
  For Q-Learning, it can be seen that classic Q-learning fails to converge as it is observed in plot from paper. Although Q-learning shows decreasing trends but even after 1 million iterations it does not fully converged. The convergence is only effect of lower learning rate of 0.001. For higher learning rate of 0.01 there is no convergence even for 1 million iterations.



  For both Foe-Q and CE-Q plots are very similar, confirming with what authors say about CE-Q learns policies that coincides exactly with Foe-Q. Although plots for both CE-Q and Foe-Q seems to be converged fully, it is much faster than plots from paper. This can be affect of hyperparameters such as epsilon decay, alpha decay. Further investigation and experimentation are needed to validate the claim.

Conclusion: Although there was stiff learning curve involved for linear programming and CE-Q algorithm, project experience was very unique and interesting experience to learn and experiment CE-Q algorithm.

The goal CE-Q is to improve design of multiagent systems (MAS). MAS designs act as central planners, equipping all agents in the system with pre-specified behaviors, but these systems are rarely compatible with agent's incentives.

References:

Dr Charles Isabel and Dr Michael Littman: Lessons: Game Theory

Amy Greenwald and Keith Hall: Correlated-Q Learning 2003

Dr. Michael Littman: Friend-or-Foe Q-learning in General-Sum Games

TA's during office hour and piazza forum.