Replicating Greenwald 2003: Correlated-Q Learning

Evan Jones
Georgia Institute of Technology College of Computing
Git Hash: ec0a728cd18a5fa96d242159b45daa824efe04b0
Georgia Institute of Technology
Atlanta, GA USA
evanjones@gatech.edu

***Abstract*—In this report we will be exploring various multi-agent Q-learning algorithms that learn equilibrium policy solution methods for Markovian games. Through replication of Greenwald & Hall 2003 [2] we will implement and discuss the resulting policy solutions of single-agent Q-learning, Friend Q-learning, Foe Q-learning, and Correlated-Q learning within the context of a grid soccer environment.**

***Index Terms*—Game Theory, Reinforcement Learning**

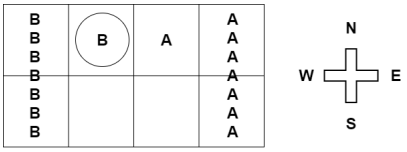

Figure 1. Soccer Environment at State "s"

I. Introduction

In recent years, the application and extension of reinforcement learning methods to classical problems in game theory has led to many novel and interesting results. The interdisciplinary approach seems natural given both fields center upon strategic agent decision making in complex environments. However, these generalizations from the typical single agent Markov decision process case (reinforcement learning) to the multi-agent multi-stage case (game theory) would not have been possible without the Markovian game framework. This framework effectively allows us to treat individual stages of a multi-stage games as their own single stage game or Markov decision process. Furthermore, work by Filar & Vrieze (1997) [6] demonstrates that individual stage equilibrium's' only persist when the multi-stage case is also in equilibrium. This profound result is what allows us to reasonably apply Q learning methods to such problems and motivates our report as we attempt to find Q functions that satisfy certain properties of different these different equilibrium solution methods.

II. Soccer Environment Overview

In this report we will be using a best efforts replication of the soccer environment specified in Greenwald & Hall (2003) [2] (Henceforth referred to as "Greenwald 2003" for brevity). The environment itself will be be a 2x4 grid that will represent our soccer field. The two agents will each occupy a single square of the field at a given time. Additionally, the ball will always be in the possession of a single player at a given time. The top and bottom squares to the immediate right and left ends of the grid (see Figure 1 the vertical columns of repeated A's and B's) will represent our goal states for which the agents will ideally attempt to enter into with possession of the ball. Agents will have five separate actions at their disposal to navigate the field $[stick, north, east, south, west]$. The execution order of player actions will be random $Bernoulli\ (1/2)$, which carries the side effect of making this a stochastic game even though the environment and actions are known and finite. Player collisions will serve as a mechanism for ball possession transfer. If in a given action sequence players are due to collide, and the player currently in possession of the ball challenges the opponents square, a steal will occur. This will result in the ball switching possession to the defending player. The game will terminate upon a player entering the any of the four goal states with possession of the ball. The corresponding agent rewards depend on if the agent scored in their goal or the opponents goal which are -100 (own-goal) and +100 respectively. Conversely, the non-scoring opponent will receive the reward opposite the scoring agent making the game zero sum. A graphical representation of the environment is provided in Greenwald (2003) [2] and is reproduced in Figure 1 with variation on the naming of goals and definition of "own-goal". In the revised representation, we implement traditional real-world soccer definitions where a players objective is to defend their goal and score in the opposing players goal. This new representation simplifies our ability to describe the environment, but will not have any further impact on the experimental results. Now that we have described a general approach to the environment's operations we will next discuss the implementation details of the game and highlight problem areas discovered during the replication. Note here that in further discussion of the environment we will refer to individual squares as numbered beginning 1-4 for the first row and ending 5-8 on the second row.

III. Soccer Implementation & Pitfalls

We call this a best efforts replication as there is quite a bit of room for ambiguity in Greenwald's specifications. Firstly, the specified environment by Greenwald fails to address how or if the valid action space of each agent changes based on their given position. As an example, consider the case where an agent is constrained by a wall above blocking its movement (ex: a player in square 2). In such a situation Greenwald

does not provide guidance on whether the agent will only be able to execute valid actions, or continue to have the full action space at its disposal with an environmental mechanism causing the agent to act randomly or remain stationary as opposed to executing an invalid action. To remedy this issue we decided that it seems logical to follow conventions used in many other grid world settings keeping the entire action space available, and using the environment to handle invalid moves by making the agent stationary. Secondly, Greenwald fails to specify if the four separate goal squares are in fact valid player defensive positions for a player without possession of the ball. In resolving this problem we decided that it seems logical to allow a player to essentially be a goalie and enter goal squares despite not having the ball. However, it is notable that in this configuration an agent defending their own goal would not be ideal as a steal in that state would result in an own-goal. Thirdly, although the game termination conditions are clear, Greenwald does not give extensive guidance on how the game is re-initialized after termination. The authors analysis of the game centers upon the given state s as depicted in the accompanying Figure 1. However, it is unclear if this state is used only in terms of error evaluation over time (given that this particular state has no deterministic pure policy) or if it also implies that this state will be repeatedly initialized. After testing both this static initialization, and a random initial configuration where players would be randomly placed in positions 2 or 6 for player B and positions 3 or 7 for player A with random possession initialization. The reason we chose only these two squares for each is to mirror the start of a real soccer match where opponents meet on opposite sides of the midfield. We found that the randomization method yielded results more consistent with Greenwald's figures and we decided to implement this approach accordingly. However, this choice came with its own set of problems when recreating and understanding how Greenwald achieved her supplied results. These issues will be further discussed in section IX. Now that the environment has been sufficiently specified we will now address the experiments and methodologies we aim to reproduce in this study.

IV. Methodology

In the following sections we will implement and conduct replication experiments for each of the four equilibrium solution methods presented in Greenwald (2003) [2] figure 3. Each of these four agents will be trained for one-million iterations on the soccer environment specified above. This interval as well as timeout conditions are implemented as per the authors specifications. Through continued interaction with the environment each agent will utilize its specific update rule and value/policy determination methods to act optimally with respect to the agent's unique objective function. The purpose of these experiments is to both evaluate the different agents' ability to converge on an a mixed equilibrium policy as well as to examine the intuition underlying these resulting policies deriving from the state s where no such deterministic policy can exist. This measure of "convergence" will take the

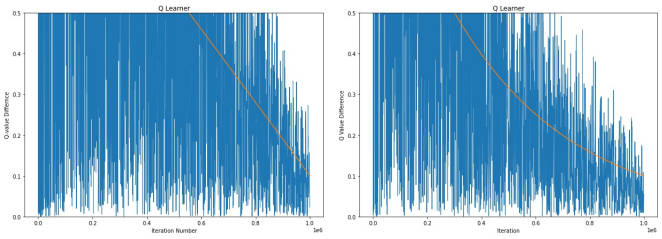

Figure 2. Q-learning Agent under Linear and Exponential Alpha Decay

following formulation: $ERR_i^t = |Q_i^t(s, \overrightarrow{a}) - Q_i^{t-1}(s, \overrightarrow{a})|$ (1), and will be updated each time state s is visited. Equation (1) measures the absolute change in a specific state action pair value and when this error is zero (over many time steps) it will indicate our given value has converged as it is no longer experiencing updates despite additional visits to the given state. Such a method is common in the multi-agent Markov game framework and provides an excellent opportunity to better understand each solution methods strengths and weakness with regard to these types of general problems. This also will give insight as to how the specification of multi-agent variants of Q learning improve upon their predecessor's shortfalls. Finally and most notably, these experiments will demonstrate how specific formulations of their objective functions allow the learners to select a matching single equilibrium in the presence of several possible equilibrium choices which prior to this optimization they were indifferent towards. In the following section we will implement the first algorithm Q-learning.

V. Q Learner

For our first experiment, we will be implementing a standard Q-learning agent and evaluating its performance in the soccer environment. Q-learning, initially introduced by Watkins [5], utilizes a tabular value representation of the state/action space which is updated incrementally as the agent explores the environment according to the equation: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t))$ (2). This method is guaranteed to converge to deterministic optimal values (and thus policies) in the single agent finite setting under easily satisfied exploration assumptions. However, in the presence of multiple agents, these convergence properties no longer hold as we will soon see empirically in our experimental results. Unlike subsequent solution methods that will follow later in this report, the Q-learner has not been adapted to the multi-agent case, and thus will be indifferent/ignorant of the other players actions. The results of the Q-learning agent are demonstrated in Figure 2 the first graph demonstrates a linearly decayed alpha value from $\alpha = 1.0 \rightarrow \alpha = 0.001$ and the second represents an exponential decay schedule with the same endpoints. The reason we supplied both graphs was to demonstrate the effect of alpha decay on the results as Greenwald neglects to provide these implementations details. Also note that these two plots did not share the same randomized seed and are both independent trials. Furthermore, due to ambiguity in the original research specifics related to the

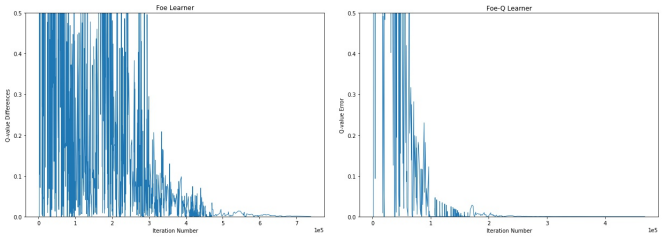

Figure 3. Foe-Q learner seeded (right) and un-seeded (left)

learners exploration policy are covered in Section IX. As we can see from both plots the Q-learner has some very large magnitude errors which in the authors y-axis limit of 0.5 tops out the plot for a large amount of time indicating no convergence whatsoever. In time we see that the oscillating of error values is reduced. However, as noted by Greenwald and made evident by our additional plot of alpha within the figure, this is not true convergence, and is only a result of our decaying learning rate alpha. In both cases of decay scheduling we can see errors directly responding to the reduction in α by forming an "envelope" around its value. This graph certainly shows that the Q-learner as we hypothesized above was unable to achieve a properly converged policy. Since this formulation of Q-learning seeks a deterministic policy for which in this state one cannot exist it will never truly converge. This reliance on pure strategies results in the agents failure, and is a key aspect of this naive approach that needs to be addressed in order to utilize these reinforcement learning methods in a multi-agent setting. In comparing our results for Q-learning to those obtained in Greenwald (2003) [2], it is evident that our results are very similar. It does seem notable to point out that Greenwald's Q-learner plot more prominently displays the envelop demonstrated in our exponential decay case. We believe that given this shape it is likely that although not explicitly stated Greenwald utilized exponential alpha decay in the initial study. Beyond this it is likely that the differences in graphs are due strictly to the difference in random initialization. Although it is evident from Greenwald's plots of Foe-Q and CE-Q that a specific random seed was used in generating the experimental results it is impossible to replicate exactly as the specific seed was never mentioned. However, given the strong semblance in results we accept that Greenwald's results were successfully replicated for this algorithm. In the next experiment, Littman (2001) [1] will address this problem directly with his introduction of Foe-Q learning.

VI. Foe Learner

In this experiment we will be implementing the Foe-Q learning algorithm set forth by Littman (2001) [1]. Under this framework Littman addresses many of the problems arising from applying direct Q-learning methods to multi-agent games by creating an expanded learning framework whereby the agent is aware of other players and their corresponding actions in the environment, and also perceives them as adversarial opponents (Foe-Q) or cooperative players (Friend-Q) (discussed in section VII). When combined, both of these extensions allow our learner to create a more realistic representation of the game, and thus develop an effective mixed strategy corresponding to the situation. Under this new framework, we will introduce an additional step within our standard Q-learning implementation. In Foe-Q, the learner will assume that the opponent is attempting to minimize its reward at all costs, given this the Foe-Q player will attempt to maximize the minimum score it hopes to achieve assuming the opponent is a true optimal adversary and play this strategy from the observed state forward. This expected value of future play replaces our value of future states in the previous Q-learning equation (2). This leads us to a standard mini-max problem in solving for the optimal policy/strategy given a reward matrix to represent this value as shown by the corresponding objective function over joint actions: $Soln(s, Q_1, Q_2) = max_{\pi \in \Pi(A_1)} min_{a_2 \in A_2} \sum_{a_1 \in A_1} \pi(a_1) Q_1(s, a_1, a_2)$ (3). Typically, we would need to have two q tables for each player to solve this problem, but given this is a zero some game the second player will always have a reward matrix corresponding to the additive inverse of the other player. Thus we only need to track a single q table in this implementation. This resulting policy will be used by our agent both in policy evaluation and state value determination. The side effect of these new extensions guarantees Foe-Q's convergence to a equilibrium solution policy. Also note that although our Foe agent will assume adversarial actions by its opponent, the opponent will in reality be following a purely random policy, but this formulation will still yield a proper solution. Due to ambiguity in the original research specifics related to the learners exploration policy are covered in Section IX. The results of the Foe-Q learner are depicted in Figure 3. We supply two plots here the first of which takes a specific random seed for a test of replication with the following CE-Q leaner in Section VIII. The second plot is simply a secondary un-seeded trial that we found converged abnormally quickly during testing. Both plots utilized exponential alpha decay over the endpoints specified by Greenwald and $\gamma = 0.9$. As we can see from the plots we achieve similar results to that of Greenwald with only random variance separating their results from the seeded plot. This small difference, like the difference in the Q-learner, is due strictly to differences in randomization that cannot be replicated exactly without a seed value. It is again notable how quickly the second un-seeded plot was able to converge and differs strongly from the results presented in Greenwald's (2003) paper. After conducting additional research into these difference we discovered a subsequent publication by the same author that revisits the 2003 paper in enhanced technical detail [4]. Interestingly enough the plots we aimed to replicate in Figure 3 of the 2003 paper are absent and instead new graphs are presented the visuals of which more strongly resemble the secondary unseeded plot of Foe-Q. It remains unclear how exactly Greenwald's approach was altered in the two papers, but we suspect in the latter paper randomized initializations were no longer applied. This would

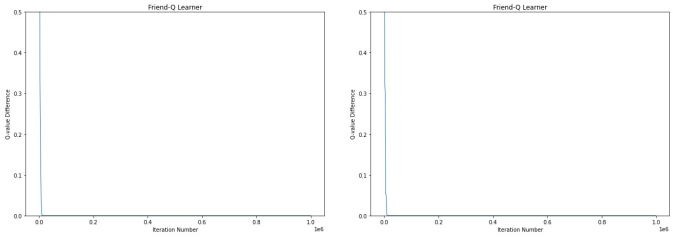

Figure 4. Friend-Q Learner seeded (left) and un-seeded (right)

allow quicker convergence as the agent would be updating at least once a game. This could so explain our results which happened to correspond with a lucky randomization (that caused the state to be visited frequently which corresponds to faster convergence w.r.t. total time-steps). Later, replication of the static initialization approach confirmed this hypothesis. Overall though, it is evident in the plot that our Foe-Q learner converges to a stationary Q value and proper mixed policy for state s far faster than could be caused by alpha decay alone as was the case in standard Q learning in either configuration. It is also notable that the Foe-Q learner achieves the same resulting final policy as the authors observed in their experiment. Next, we will explore the other half of Littman's paper through Foe-Q's alter ego friend-Q.

VII. Friend Learner

In this experiment we will be implementing the alternative formulation of the Foe Q learner demonstrated by Littman (2001) [2] whereby instead of treating other players as adversaries we will instead treat them as if they are cooperative players friendly to our agent's goals. This formulation will accordingly replace the maxi-min linear program used in Foe-Q with a standard max operator over actions as demonstrated in the following objective function over joint actions: $Soln(s, Q_1, Q_2) = max_{a_1 \in A_1, a_2 \in A_2} Q_1(s, a_1, a_2)$ (4). Aside from these differences, the two implementations are fundamentally equivalent. Just as in the prior experiments, the agent will play with a random agent which although not adversarial is not exactly friendly either. For this reason behavior by the Friend agent may fallaciously expect cooperative action, but will receive mixed responses from the random agent. For the friend learner, we used the same epsilon, alpha, and gamma schedules and parameters utilized in the prior foe experiment over one-million iterations plotting absolute value of q-value differences for each update according to (1). Again due to ambiguity in Greenwald specification of exploration we will save further discussion for this for Section IX. The results are demonstrated in Figure 4 in both seeded and un-seed trials. As we can see from the plots, the Friend Q learner converges to a policy the quickest among all learners. However, as we discussed above it's policies are naive in the assumption of cooperation and thus are not rational. We note here that the graphs when compared to those presented in Greenwald (2003) [2] are vastly similar, and we note the convergence to the same policies/behavior for player B in state s as observed by the author. Any difference in plots is due strictly to difference in random seeding. The analysis of this Friend-Q approach would be better suited under an truly cooperative environment and preferably general sum game. Such an alternative environment would highlight the unique behavior and novel properties of this solution method far better than possible in this example. However, as a whole we can conclude that we were able to successfully replicate the Friend-Q leaner results presented by Greenwald's experiments. Next we move onto our final solution method correlated equilibrium.

VIII. Correlated-Q Learner

In our final experiment we will explore the algorithm known as correlated-q learning which was initially introduced in Greenwald (2003) [2] based on the correlated equilibrium solution method. The primary intuition into this solution method can be best explained through a simple example of traffic lights. Traffic lights allow drivers to correlate their behaviors at intersections so as to avoid bad outcomes such as crashes. For that reason traffic lights should never have green showing on both directions, and in the same general premise, the correlated-Q equilibrium solution method aims at such a compromising outcome in the general case of Markov games. Unlike our past algorithms, Correlated-Q will rely upon 2 distinct Q tables one for each agent in its computation which takes the following form as an objective function:$\sigma \in max_{\sigma \in CE} \sum_{i \in I} \sum_{\vec{a} \in A} \sigma(\vec{a}) Q(s, \vec{a})$ (5). As we can see from the objective above this solution method will return the equilibrium policy that maximizes the total utility of both players combined rewards and for this reason is also dubbed a utilitarian formulation of the correlated equilibrium solution method. In a general sum Markov game this correlation among agents in their joint actions yields utilitarian outcomes. However, as noted above, our testing environment is a zero sum game so this coordination of action will not improve overall combined utility which will remain fixed at 0. For this reason in the zero sum Markov game, the Correlated-Q learner will in-fact mirror the behavior of the foe-Q learner. Given this insight our experimental results for correlated-Q should bear similarity to those produced in our above Foe-Q results if the randomized seeds correspond. Similar to past implementations we followed the same approach in setting model hyper-parameters and decay schedules. The CE-Q results are demonstrated in Figure 5 in both seeded and un-seed trials. The seeded plot for CE-Q utilizes the same seed used in the Foe-Q left side plot so as to test if Foe-Q and CE-Q follow the same updates as theory would suggest. As we can see the first of our reproduced graphs for Correlated-Q directly mirrors the results of the first reproduced Foe-Q graph in Figure 5. This is because for both of these initialization we seeded the random number generators equivalently to see if under identical operating conditions they produce the same updates and results as demonstrated in Greenwald's identical plots in figure 3 for Foe-Q and CE-Q. We can also see that generally speaking, both the seeded and un-seed plots strongly resemble the results of CE-Q shown in Greenwald's Figure 3.

As in previous experiments it is evident that the only difference is due to different random seeds, and thus we can safely conclude that we were able to successfully replicate each of Greenwald's experiments.

IX. Algorithmic Implementation Notes & Pitfalls

Although we were able to achieve similar results for each of the figures in Greenwald's Figure 3 [2], this was far from a straight forward. Despite the paper's detailed discussion of theory it leaves little to no details to go on in practically replicating or implementing the outlined algorithms, environments, and experimental results. In addition to the unique pitfalls addressed in each algorithmic implementation in the sections above, we were faced with four other primary difficulties in replicating the paper. Firstly, although their minimum values were specified, Greenwald leaves out key details on decay schedules for the alpha parameter. This parameter and decay approach is crucial to any such iterative learning algorithm, and provides key context to the results displayed in Figure 3 (as discussed in Section V). This is especially important in the case of Q-learning where the author directly relates the graph's behavior to the alpha "envelope", but fails to support this assertion with a corresponding alpha schedule that would add supporting evidence to this argument. Secondly, although replicated it is very unclear how exactly the author arrives at the graphs depicted in Figure 3, and why these resulting graphs differ in subsequent publications such as the technical supplements published in 2007 [4]. As we noted in Section VI, the graphs supplied in the technical supplement are far more sparse and converge much quicker than those presented in the 2003 paper. Given that similar descriptions on implementation were supplied in both papers it is strange to see such divergence in results. It seems that clarification by the author on the initialization method and update frequency could have greatly helped this situation. Additionally, it is unclear why the author decided to cut the y axis in this fashion as it gives researchers even less vision into the underlying data especially early on in iteration when q-values are high. Thirdly, during the implementation of each of the solution methods requiring linear programming considerable time was spent deriving and verifying the proper constraints to place upon the particular problem. Although as the author I'll admit to not being especially keen on linear programming methods or solvers, but there could've been significantly more information on the implementation and exact solvers used in Greenwald's report. Numerical stability issues can arise with fast solvers like GLPK so if the author used this solver explaining how they handled such issues is also crucial to replication. On the other hand, if the author used more flexible methods like cone-lp (which we did not experience issues in related to numerical stability) this should have been noted as the run-time seemed extremely longer than would be expected. Finally, the author mentions using an on/off policy approach to action selection throughout the paper via values of epsilon corresponding to $\epsilon_{on-policy} = .001$ and $\epsilon_{off-policy} = 1.0$. However, these definitions directly conflict with the traditional

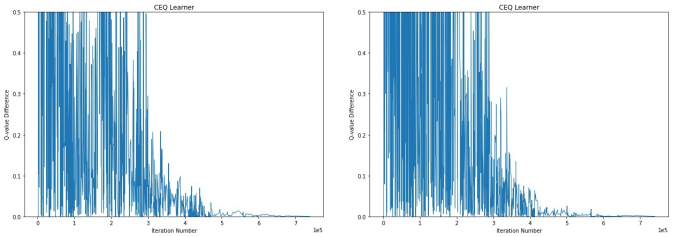

Figure 5. CE-Q Learner seeded (left) and un-seeded (right)

definition of on/off policy learning set forward by Sutton [3]. Under the framework of Sutton, both of Greenwald's methods would be considered off policy, the prior of which is nearly on policy, but not quite as it still has some amount of randomness. This also adds further confusion to the treatment of epsilon in our algorithmic implementations as we were unclear as to whether this means that epsilon should take only these discrete values and policy approaches or if we should apply traditional reinforcement learning methods with an e-greedy action selection over a decaying epsilon schedule. In the end the only approach that yielded similar results to Greenwald to have a static epsilon value with a switch point from Greenwald's definition of off policy to on policy three quarters of the way through training. Specific guidance on the selection function is crucial to replication and is probably the single most important area of further clarification needed for replicating of Greenwald's results.

X. Discussion & Closing Thoughts

Collectively, this project was immensely difficult not due to the difficulty of the underlying theory or algorithms, but exclusively due to the opaqueness by which the authors of these various papers demonstrate their experimental results. Although detailed notes on implementation may seem rather trivial, they can often be the key difference in other researcher being able to replicate in one hour vs. one week. Additionally, although the results and usage of these reinforcement learning methods in the context of game theory is interesting, it seems that the methods are rather impractical for any real or large problems. However, despite these concerns it is enlightening to understand just how knowledge and breakthrough in fields like reinforcement learning can trickle down and disrupt other fields into new interesting results to classic problems.

References

[1] Littman, Michael L. "Friend-or-foe Q-learning in general-sum games." In ICML, vol. 1, pp. 322-328. 2001.
[2] Greenwald, Amy, Keith Hall, and Roberto Serrano. "Correlated Q-learning." In ICML, vol. 3, pp. 242-249. 2003.
[3] Andrew G. Barto and Richard S. Sutton "Reinforcement Learning: An Introduction" Second Edition, The MIT Press, 2020.
[4] Greenwald, Amy, Keith Hall, and Martin Zinkevich "Correlated Q-Learning Technical Supplement". Journal of Machine Learning Research I. 2007, 1-1.
[5] Watkins C. "Q-Learning" Machine Learning, 8, 279-292. Boston: MA. Kluwer Academic Publishers April 1992.
[6] Filar, J., & Vrieze, K. (1997) "Competitive Markov Decision Processes". Springer-Verlag.