CSCE475/875 Multiagent Systems

Handout 12: Game Day 1 Learning Day Analysis

September 22, 2017

State Transition Map and Rewards

There were six states (S1-S6) and six actions (A1-A6). Each team started with S1. Each team was capable of performing all six actions. Table 1 shows the rewards for transitioning into each state and, its average and standard deviation, based on a Gaussian distribution.

	Average	Std. Dev.
S1	$0	$10
S2	$100	$10
S3	$1500	$10
S4	$500	$10
S5	$5000	$10
S6	$1000	$10

Table 1. Rewards, average and standard deviation values, Gaussian distribution.

Table 2 shows the probabilistic transition map for each state-action pair. Looking at both Tables, if one aimed to obtain the highest reward for a state (i.e., S5 @ $5000), then starting for S1, one would probably have to go with A4 to transition into S4 (with a high probability @ .7), and then go with A5 to transition into S5 (with a high probability @ .9). And then to get back to S1, one could perform an action of A6, if so desired. This sequence of A4-A5-A6, when repeated, should allow an agent to reach S5 with a relatively high probability (= .7 x .9 x .5 = .315), and a relatively high reward (= $500 + $5000 + $0 = $5500). With enough exploration, an agent should be able to discover this sequence.

	A1	A2	A3	A4	A5	A6
S1	→ S1 (.50) → S2 (.30) → S3 (.15) → S4 (.05)	→ S2 (.80) → S3 (.20)	→ S2 (.10) → S3 (.90)	→ S1 (.05) → S2 (.25) → S4 (.70)	NA	NA
S2	→ S1 (.70) → S2 (.15) → S4 (.15)	→ S1 (.55) → S3 (.35) → S4 (.10)	NA	NA	NA	→ S5 (.50) → S6 (.50)
S3	NA	NA	→ S1 (.70) → S2 (.20) → S4 (.10)	→ S1 (.60) → S3 (.30) → S4 (.10)	NA	NA
S4	→ S1 (.60) → S2 (.20) → S3 (.20)	→ S1 (.65) → S2 (.14) → S3 (.20) → S4 (.01)	→ S1 (.98) → S2 (.02)	NA	→ S5 (.90) → S6 (.10)	NA
S5	NA	NA	NA	NA	NA	→ S1 (.50) → S2 (.30) → S3 (.15) → S4 (.05)
S6	→ S1 (1.0)	→ S2 (1.0)	→ S3 (1.0)	→ S4 (1.0)	NA	NA

Table 2. State transitions by actions. NA for a state-action cell means the action is not applicable for the state.

Because of the limited time on Game Day, we did not expect teams to obtain *accurate* Q-values. However, teams should be able to obtain *fairly accurate ordering* of their Q-values. The ordering of the best state-action pairs (Q(s,a)) is as follows:

Group 1: (S5,A6)
Group 2: (S4,A5); (S5,A1); (S5,A2); (S5,A3); (S5,A4); (S5,A5)
Group 3: (S2,A6); (S6,A4)
Group 4: (S3,A3); (S3,A4); (S4,A4); (S6,A2); (S6,A3)
Group 5: (S1,A4); (S4,A1); (S4,A2); (S4,A6); (S6,A1); (S6,A5); (S6,A6)

For the above, we also define a function called Group_true(s,a) that returns the group ID of a state-action pair. So, for example, Group_true(S5,A6) is 1; Group_true(S4,A6) is 2; Group_true(S5,A1) is 2; and so on.

Team Statistics

Tables 3 and 4 show the ordering of the teams after Round 1 and Round 2, respectively.

To compute the accuracy of a Q-table, we use the grouping shown earlier. We consider only the top 18 state/action pairs in each team's Q-table (where 18 is half of the 36 possible values). (Important Note: the last group actually only has 4 elements (not 7) when we limit ourselves to only looking at the top 18 for each group. We however put 7 state/action pairs in Group 5 to be fair to teams since they are all pretty equivalent in that group, and using only 4 would mean teams wouldn't get credit if they had the other 3 (equivalent) pairs, instead.)

First, we sort each team's Q-values.

And second, for each state-action pair on the sorted list, we assign Group_found(s,a) using the 1-6-2-5-7 grouping strategy. So, take Dishonest Agent's Round 1 ordering: Group_found(S2,A6) is 1; Group_found(S6,A2) is 2; Group_found(S6,A3) is 2; and forth. (Please see the color-coding in Tables 3 and 4).

Third, we compute two subvalues: matching score, and non-matching score. For matching score, if Group_found(s,a) == Group_true(s,a), then we will multiply it with a weight and add it to the score: weights = 1, 0.5, 0.25, 0.125, and 0.1 for the five groups, respectively. This scheme rewards teams that have high accuracy for the top state-action pairs. For the non-matching score, if Group_found(s,a) – Group_true(s,a) == 1 OR Group_true(s,a) – Group_found(s,a) == 1, then we will multiply it with the lower group weight of Group_found(s,a), Group_true(s,a) and add to the score. This is to compensate state-action pairs that miss their true grouping just by one group.

Then we add up the matching and non-matching scores.

Rank	Dishonest Agents	Quiero MAS	The Whales	Rogue Wan	Git Rekt	Winter Slayers	Team Cerberus	Simulated Ground Truth
1	S2-A6	S4-A5	S4-A5	S2-A6	S2-A6	S1-A4	S4-A5	**S5-A6**
2	S6-A2	S2-A2	S4-A4	S1-A3	S2-A5	S3-A4	S2-A2	**S4-A5**
3	S6-A3	S3-A4	S1-A4	S5-A6	S3-A2	S3-A1	S5-A6	**S5-A1**
4	S1-A3	S1-A4	S5-A6	S3-A4	S3-A3	S4-A6	S1-A3	**S5-A2**
5	S5-A6	S4-A2	S4-A1	S1-A6	S2-A1	S4-A3	S6-A3	**S5-A3**
6	S3-A4	S1-A1	S3-A3	S6-A1	S2-A2	S4-A4	S2-A6	**S5-A4**
7	S4-A3	S1-A3	S6-A1	S2-A2	S1-A4	S1-A2	S3-A4	**S5-A5**
8	S1-A2	S2-A6	S1-A2	S1-A5	S1-A1	S1-A3	S3-A3	**S2-A6**
9	S4-A2	S6-A1	S2-A1	S2-A1	S1-A2	S1-A1	S3-A1	**S6-A4**
10	S6-A1	S5-A6	S2-A5	S4-A3	S1-A3	S1-A5	S3-A2	**S3-A3**
11	S3-A3	S1-A5	S1-A1	S3-A3	S1-A5	S1-A6	S3-A5	**S3-A4**
12	S4-A1	S1-A6	S1-A3	S1-A4	S1-A6	S2-A2	S1-A4	**S4-A4**
13	S1-A4	S4-A1	S1-A5	S4-A1	S2-A3	S4-A1	S2-A3	**S6-A2**
14	S1-A1	S4-A3	S1-A6	S2-A5	S2-A4	S4-A2	S3-A6	**S6-A3**
15	S2-A2	S1-A2	S2-A2	S2-A4	S3-A1	S3-A3	S1-A2	**S1-A4, S4-A1, S4-A2, S4-A6, S6-A1, S6-A5, S6-A6**
16	S2-A1	S3-A3	S2-A3	S2-A3	S3-A4	S2-A1	S1-A1	
17	S1-A5	S2-A1	S2-A4	S3-A6	S3-A5	S2-A3	S2-A5	
18	S1-A6	S2-A4	S2-A6	S3-A5	S3-A6	S2-A4	S2-A4	

Table 3. The ordering of state-action pairs from each team after Round 1. (Only the top 18 state-action pairs are listed) Colors show grouping.

Rank	Dishonest Agents	Quiero MAS	The Whales	Rogue Wan	Git Rekt	Winter Slayers	Team Cerberus	Simulated Ground Truth
1	S2-A6	S4-A5	S1-A4	S4-A5	S2-A6	S5-A6	S4-A5	**S5-A6**
2	S6-A2	S1-A4	S4-A5	S1-A4	S4-A5	S1-A3	S2-A2	**S4-A5**
3	S2-A2	S2-A2	S5-A6	S2-A6	S5-A6	S3-A4	S1-A3	**S5-A1**
4	S1-A3	S3-A4	S6-A1	S5-A6	S1-A3	S2-A2	S1-A1	**S5-A2**
5	S5-A6	S5-A6	S4-A4	S6-A1	S3-A4	S4-A5	S3-A3	**S5-A3**
6	S6-A3	S6-A1	S3-A3	S3-A4	S6-A3	S3-A1	S3-A4	**S5-A4**
7	S3-A4	S4-A2	S2-A1	S1-A3	S6-A4	S1-A4	S3-A6	**S5-A5**
8	S4-A3	S1-A1	S1-A2	S4-A6	S1-A4	S4-A2	S5-A6	**S2-A6**
9	S1-A2	S1-A3	S5-A4	S1-A6	S2-A1	S1-A2	S3-A1	**S6-A4**
10	S4-A2	S2-A6	S4-A1	S2-A4	S4-A1	S3-A3	S2-A3	**S3-A3**
11	S6-A1	S1-A5	S2-A5	S4-A4	S1-A1	S1-A5	S4-A3	**S3-A4**
12	S3-A3	S1-A6	S1-A1	S2-A2	S1-A2	S1-A6	S4-A4	**S4-A4**
13	S4-A1	S4-A1	S1-A3	S1-A5	S1-A5	S4-A1	S3-A2	**S6-A2**
14	S1-A4	S4-A3	S1-A5	S2-A1	S1-A6	S4-A4	S6-A3	**S6-A3**
15	S1-A1	S1-A2	S1-A6	S4-A3	S2-A3	S4-A6	S5-A2	**S1-A4, S4-A1, S4-A2, S4-A6, S6-A1, S6-A5, S6-A6**
16	S2-A1	S3-A3	S2-A2	S3-A3	S2-A4	S3-A2	S2-A6	
17	S1-A5	S2-A1	S2-A3	S4-A1	S2-A5	S3-A5	S3-A5	
18	S1-A6	S2-A4	S2-A4	S2-A5	S3-A1	S3-A6	S2-A4	

Table 4. The ordering of state-action pairs from each team after Round 2. (Only the top 18 state-action pairs are listed) Colors show grouping.

Now, we present the more detailed team statistics in Tables 5-7. The number of transactions and rewards were tallied based on the log that our program captured during the Game Day. As shown in Table 5, 3 teams did better than the average, and 4 teams performed below. Note also that Team Cerebrus, with only 18 actions (the fewest), obtained the best order accuracy. (**Note: This is counter-intuitive!)** Dishonest Agents performed the most actions (63), earning the largest amount of rewards ($46,233.10). The Whales

Team Name	#trans	Rewards	Efficiency	Normalized	Order Accuracy	Normalized	Total
Dishonest Agents	**63**	**$46,233.10**	$733.86	**1**	0.925	0.627	**1.627**
Quiero MAS	53	$23,717.94	$447.51	0.513	0.95	0.644	1.157
The Whales	47	$40,215.84	**$855.66**	0.87	1	0.678	1.548
Rogue Wan	40	$23,731.13	$593.28	0.513	0.825	0.559	1.073
Git Rekt	24	$10,371.00	$432.13	0.224	0.1	0.068	0.292
Winter Slayers	22	$15,027.37	$683.06	0.325	0.3	0.203	0.528
Team Cerberus	18	$11,969.04	$664.95	0.259	**1.475**	**1**	1.259
Average	**38.14**	**$24,466.49**	**$630.06**	**0.529**	**0.796**	**0.54**	**1.069**

Table 5. *Statistics of Round 1*. Dishonest Agents had the best total score, balancing between rewards and order accuracy, for Round 1. Team Cerberus scored the highest order accuracy with 1.475, while Dishonest Agents obtained the largest amount of rewards with $46,233.10. Bold red texts = high value

Table 6(a) shows only the statistics during Round 2, and *not* the total. There were on average more transactions in Round 2 compared to those in Round 1 (more than twice larger: 91.29 vs. 38.14) even though Round 2 was only 5 minutes longer (25 minutes vs. 20 minutes). In terms of Rewards, as expected, Round 2 yielded a higher average than Round 1 ($122,687.60 vs. $24,466.49). This was due to two factors. First, each team's operation, on average, was smoother in Round 2. Second, all teams exploited to gain rewards more efficiently ($1,291.29 per transaction vs. $630.06 per transaction). The average order accuracy for Round 2 was higher than that for Round 1 (1.332 vs. 0.796) as expected due to teams carrying out more actions and gaining more "learning episodes." Note also that Winter Slayers' order accuracy went from 0.3 in Round 1 to 1.95 in Round 2!

Team Name	#trans	Rewards	Efficiency	Normalized	Order Accuracy	Normalized	Total
Dishonest Agents	**168**	$204,244.47	$1,215.74	0.987	1.025	0.526	1.513
Quiero MAS	114	**$206,843.26**	**$1,814.41**	**1**	1.325	0.679	**1.679**
The Whales	85	$136,353.07	$1,604.15	0.659	1.35	0.692	1.352
Rogue Wan	85	$149,950.54	$1,764.12	0.725	1.575	0.808	1.533
Git Rekt	63	$63,012.31	$1,000.20	0.305	1.35	0.692	0.997
Winter Slayers	54	$55,441.92	$1,026.70	0.268	**1.95**	**1**	1.268
Team Cerberus	70	$42,960.66	$613.72	0.208	0.75	0.385	0.592
Average	**91.29**	**$122,687.60**	**$1,291.29**	**0.593**	**1.332**	**0.683**	**1.276**

Table 6(a). *Statistics of Round 2 (**not** including Round 1's rewards and # transactions)*. Quiero MAS had the best total score, balancing between rewards and order accuracy, for Round 2. Winter Slayers scored the highest order accuracy with 1.95 while Dishonest Agents obtained the largest amount of rewards with $206,843.26. Bold red texts = high value

Furthermore, though the grand total of the two rounds was not used in our scoring directly, we provide the grand total values for all teams here as a reference in Table 7. Dishonest Agents performed the most actions in each round. However, they did not exploit as well as Quiero MAS in Round 2.

Team Name	#trans	Rewards 1	#trans	Rewards 2	#trans Total	Rewards Total
Dishonest Agents	**63**	**$46,233.10**	**168**	$204,244.47	**231**	**$250,477.57**
Quiero MAS	53	$23,717.94	114	**$206,843.26**	167	$230,561.20
The Whales	47	$40,215.84	85	$136,353.07	132	$176,568.91
Rogue Wan	40	$23,731.13	85	$149,950.54	125	$173,681.67
Git Rekt	24	$10,371.00	63	$63,012.31	87	$73,383.31
Winter Slayers	22	$15,027.37	54	$55,441.92	76	$70,469.29
Team Cerberus	18	$11,969.04	70	$42,960.66	88	$54,929.70
Average	**38.14**	**$24,466.49**	**91.29**	**$122,686.60**	**129.43**	**$147,153.09**

Table 7. *Total rewards and total number of transactions after Round 2*. Dishonest Agents had the highest rewards total with $250,477.57. Bold red texts = high value

To compute the final score for the Learning Day, we compute the following score for each round:

$$Score = OrderAccuracyNormalized + RewardsNormalized$$

And then we combine both rounds of scores to obtain the final score:

$$FinalScore = 0.4*Score(Round1) + 0.6*Score(Round2)$$

For *OrderAccuracyNormalized*, we normalize each team's order accuracy by the best order accuracy achieved by a team. So, the best team will have its *OrderAccuracyNormalized* = 1.0.

For *RewardsNormalized*, we normalize each team's total rewards (i.e., rewards earned from performing actions + revenue from selling Q-table – cost from purchasing Q-table) with the best rewards earned by a team. So, the best team will have its *RewardsNormalized* = 1.0.

Table 8 shows the result. Overall, Dishonest Agents scored the highest overall total with 1.559. Quiero MAS placed second, followed closely by The Whales and Rogue Wan. Winter Slayers placed fifth. Team Cerberus and Git Rekt finished as sixth and seventh, respectively. Note that two teams significantly improved their scores from Round 1 to Round 2: Winter Slayers, Git Rekt. Rogue Wan and Queiro MAS also improved considerably. The Whales and Dishonest Agents dropped a bit. Team Cerberus dropped significantly from 1.259 to 0.592.

Team	Round 1 Score	Round 2 Score	Final Game Day Score
Dishonest Agents	**1.627**	1.513	**1.559**
Quiero MAS	1.157	**1.679**	1.470
The Whales	1.548	1.352	1.430
Rogue Wan	1.073	1.532	1.348
Git Rekt	0.292	0.997	0.715
Winter Slayers	0.528	1.268	0.972
Team Cerberus	1.259	0.592	0.859

Table 8. *Final Game Day scores*. Final Game Day Score = 0.4*Round 1 Score + 0.6*Round 2 Score. Bold text = high value.

Individual Team Analysis

First, Table 9 shows the learning rate and discount factor used in Round 1 and Round 2 by each team.

Team Name	Round 1		Round 2	
	Alpha	Beta	Alpha	Beta
Dishonest Agents	Function of t	Function of t	Function of t	Function of t
Quiero MAS	0.3	0.6	0.1	0.95
The Whales	0.7	0.3	0.7	0.5
Rogue WAN	Max(.8-0.002n,0.1)	Min(.4+.002n,0.9)	0.3	0.8
Git Rekt	0.75	0.9	0.7	0.9
Winter Slayers	0.9	0.9	0.8	0.8
Team Cerberus	0.75	0.33	0.33	0.85

Table 9. Learning rates and discount factors used by each team for Round 1 and Round 2.

Before we start looking at teams individually, here is a general sense of the two rounds and the role of the intermission's information sharing.

In general, Round 1 is more for exploration, and Round 2 is for a bit more exploitation. That is, Round 1 should be used to explore different state-action pairs. And as a result, one should use a higher learning rate, to emphasize each current transaction and its reward more. If a team carried out a large number of actions in Round 1, then that team could use Round 2 more for exploitation since it would be rather confident that its Q-values had converged. In that scenario, using a lower learning rate and a bigger discount factor would help towards that.

There are also other factors. Note that for any learning approach to work, in particular for reinforcement learning to work, there must be sufficient learning episodes. In this Game Day, that means each team should secure a lot of transactions in order to better model the stochastic nature of the environment.

Table 10 documents my comments on each team's worksheet and reports. My observations are contextualized on the discussions above. For "Post-Game", I selected some statements from each team's post-game analysis.

Team Name	Comments	
The Whales	**Pre-Game**	Lack of understanding of alpha and beta. Started with a low alpha = 0.7 and an even lower beta = 0.3. Should be much higher for both. Rationales were not quite clear. Used a program and the "reducing epsilon-greedy" algorithm, but didn't cite proper reference for this algorithm. Also, not sure how epsilon was related to alpha and beta.
	Round 1 Tracking	Missing V(s) values. Missing Q(s,a) updates.
	Mid-Game	Pregame strategy did not match actions during intermission. Didn't change alpha, changed beta but without justification.
	Round 2 Tracking	Missing V(s) values.
	Post-Game	Their program did not output the Q(s,a) and V(s) values so they were not able to record them during game time.
	My Observation	This team did fairly well, using an action selection algorithm.
Git Rekt	**Pre-Game**	Lack of understanding of alpha. How to set its value is not whether the environment is more deterministic or more stochastic. It is more about whether an agent knows more about its environment or not. An

		environment can be very stochastic, yet, if the agent knows very much about the environment, then alpha should be low. Lack of understanding of beta. No strategy for mid-game and Round 2.
	Round 1 Tracking	No updated Q(s,a), no updated V(s).
	Mid-Game	Lack of rationales for decisions made: e.g., why changed alpha from .75 to .70? Mid-game observations were not clear. Note that they avoided state-action pairs that led to 0 rewards.
	Round 2 Tracking	No updated Q(s,a), no updated V(s).
	Post-Game	Lack of rationales for observations made.
	My Observation	This team performed almost 3x as many actions in Round 2 than in Round 1. Not quite well prepared.
Team Cerberus	**Pre-Game**	Lack of understanding on beta: a low beta would delay learning convergence, especially if the optimal state or state-action pairs are surrounded by layers of bad state or state-action pairs.
	Round 1 Tracking	Complete. Accurate.
	Mid-Game	"We will increase Beta and decrease Alpha to increase reward and decrease learning. We will be looking to exploit the Q-table to get the maximum reward." Did not replace Q-table because they felt that they covered a large range of Q values.
	Round 2 Tracking	Complete. Accurate.
	Post-Game	"The more trials we did, the higher our Beta should have gone. A good selection at one time wasn't necessarily a good selection at another time. The more trials [*learning episodes*] the algorithm had, the closer the probabilities got to the actual values."
	My Observation	This team was able to generate the highest order-accurate Q-table in Round 1. But their accuracy dropped in Round 2. Their high accuracy in Round 1 could be due to (1) their spreading actions across the action space, and (2) the stochastic environment. Their main issue was the lack of learning episodes.
Winter Slayers	**Pre-Game**	Ran simulations to identify learning rate and beta value. Simulation results might not be too appropriate. No strategy on how to select actions. No mid-game strategy. No Round 2 strategy. No discussions on exploration vs. exploitation.
	Round 1 Tracking	Q(s,a) and V(s) not correctly updated. Incomplete
	Mid-Game	Changed alpha and beta from 0.9 to 0.8, for both. No rationales were given. Provided some observations specific to state-action pairs. But no observations about own performance and whether to change strategy.
	Round 2 Tracking	Q(s,a) and V(s) not updated. Incomplete
	Post-Game	They observed that in Round 2 the same state-action pair resulted in negative rewards consistently. No high-level insights or observations.
	My Observation	This team executed fairly well in Round 2 after a slow start. They improved vastly from Round 1 to Round 2.
Dishonest Agents	**Pre-Game**	This team built a visualization system to help them compute and tabulate Q values, generate tracked data, and so forth. Very prepared. "Our overall strategy is to train our Qtable quickly, exploiting the benefits of being able to input relatively large amounts of data points quickly." They met their objective. They performed 231 actions (largest). Very good understanding of the alpha and beta, and the exploration vs. exploitation tradeoff. Use an epsilon to randomly choose some actions. Well rationalized. Thoughtful strategy that covered all facets of the game day.
	Round 1	Complete.

	Tracking	
	Mid-Game	No mid game observations.
	Round 2 Tracking	Complete.
	Post-Game	"Do random stuff to explore more". Lack of details and insights.
	My Observation	Vey well prepared and organized. They performed the largest numbers of actions in each round. They balanced the exploration vs. exploitation tradeoff fairly well.
Quiero MAS	**Pre-Game**	"Our high-level strategy is to focus on learning in round 1 and then focus on exploiting the knowledge we've collected to maximize rewards in round 2." Lack of understanding of alpha. Alpha should be higher during exploration, not lower. Had a helper program. Had an action selection strategy for both rounds.
	Round 1 Tracking	None.
	Mid-Game	Rationales for lowering alpha from 0.3 to 0.1 were not quite supported by evidence in Q-table.
	Round 2 Tracking	None.
	Post-Game	"Our good results proved the value of our overall strategy." But actually, the team were able to gain a lot of rewards due to them not performing ANY tracking during both rounds. Other teams practiced tracking.
	My Observation	The team did very well in Round 2, moving from 4th in Round 1 to 1st in Round 2. They did this by being very efficient: gaining largest amount of rewards on average per action in Round 2.
Rogue WAN	**Pre-Game**	Used a function to determine alpha and another to determine beta. But beta might be too low. Provided strategy for exploitation and exploration. Used breadth-first search strategy to explore. Used a program.
	Round 1 Tracking	Complete.
	Mid-Game	"Probably a high learning rate would be even better". "Most teams are confident in their Q-values, so it doesn't seem like there are going to be any interactions." Rationales provided for changes in alpha and beta.
	Round 2 Tracking	Complete.
	Post-Game	Made very good observation about their exploration vs. exploitation strategy. "We can observe that the Q-values with relatively low scores in the first round did not change too much." Pointed out the risk of choosing a particular action for its immediate rewards.
	My Observation	The team did very well in Round 2, moving from 5th in Round 1 to 2nd in Round 2. Their Q-table at the end of Round 2 was ranked 2nd in order accuracy. Thus, though they exploited, they also improved their Q-table.

Table 10. My comments and observations of team strategies, worksheets, and reports.

Team	Round 1 Score	Round 2 Score	Final Game Day Score
Dishonest Agents	**1.627**	1.513	**1.559**
Quiero MAS	1.157	**1.679**	1.470
The Whales	1.548	1.352	1.430
Rogue Wan	1.073	1.532	1.348
Git Rekt	0.292	0.997	0.715
Winter Slayers	0.528	1.268	0.972
Team Cerberus	1.259	0.592	0.859

Lessons Learned

Here are some overall lessons learned.

1. In general, more transactions led to better learning. Thus, acting quickly and efficiently was critical. Teams that were slow in submitting their actions received fewer transactions, leading to poorer performances.
2. Using a low learning rate in Round 1 usually did not fare well. Using a low discount factor also did not yield accurate Q-values.
3. Lowering the learning rate or keeping it the same appeared to work better than increasing the learning rate from Round 1 to Round 2 for this MAS environment. In general, ***increasing the learning rate as time progresses would tend to unlearn what has been learned.***
4. Using a high discount factor could have a clamping effect on the learning performance brought on by a high learning rate. This is because looking into the future term essentially incorporates other Q-values into the fray.
5. Note also that Team Cerberus, with only 18 transactions (or actions) in Round 1, was able to obtain very high order accuracy. That indicated that the number of transactions, while important in obtaining accurate Q-values, the range of actions taken also played a role.
6. Several teams pointed out the nature of a tradeoff at play: ***trying to maximize rewards while trying to maximize the order accuracy***. These two objectives are in a tug-of-war. Maximizing rewards reduces exploration and increases exploitation, and vice versa with maximizing the order accuracy. Several teams had adopted an opportunistic balancing act: if they encountered a "rewarding" good state, they would keep acting on it until it transitioned out.
7. Teams that were better prepared—that came with the iterative valuation of the Q-learning algorithm and/or a program/application—performed better and thus were ranked higher. As an agent, each team should be observant, adaptive, responsive, and reflective. Not all teams were "responsive" in a timely manner.
8. **Note also that the Q-learning or reinforcement learning does *not* tell us which actions to take given a particular state. However, it does *inform* us that up to now, based on our experience, the Q-value of some state-action pairs and the value of a state. This information allows us to carry out our decision making: Should we explore some more? Should we exploit now?**

Game Days League

Here are the League Standings.

Team Name	Learning Day	Voting Day	Auction Day	League Standings
Dishonest Agents	1			1
Quiero MAS	2			2
The Whales	3			3
Rogue Wan	4			4
Winter Slayers	5			5
Team Cerberus	6			6
Git Rekt	7			7

Addendum

We ran hundreds of thousands of iterations given Tables 1 and 2, with different alpha (learning rate) and beta (discount rate) values, to generate the Q-tables. Here we include a table for beta = 0.8 to give you a sense of the Q-value for each state-action pair.

S5	a6	10786.2249
S4	a5	8916.7527
S5	a1	8628.9786
S5	a2	8628.9786
S5	a3	8628.9786
S5	a4	8628.9786
S5	a5	8628.9786
S6	a4	8133.4008
S2	a6	7667.849
S3	a3	7247.5822
S3	a4	7216.2336
S6	a2	7134.2779
S4	a4	7133.4008
S4	a6	7133.4008
S1	a4	6798.9062
S6	a3	6798.0645
S6	a5	6506.7194
S6	a6	6506.7194
S6	a1	6439.1237
S4	a1	6149.9427
S2	a3	6134.2779
S2	a4	6134.2779
S2	a5	6134.2779
S4	a2	6125.1762
S1	a2	6067.0352
S4	a3	5953.0268
S2	a1	5897.5384
S2	a2	5834.1807
S1	a3	5831.6858
S3	a1	5798.0645
S3	a2	5798.0645
S3	a5	5798.0645
S3	a6	5798.0645
S1	a1	5786.2249
S1	a5	5439.1237
S1	a6	5439.1237