Machine Learning Project 2018-2019: Report 2

Hendrik Serruys, Emile Breyne, Andreas Stieglitz

May 2019

1 Task 2: Opponent modelling

In Task 2, we use the zero-sum game of Rock-Paper-Scissors (RPS) to investigate the value of Opponent Modelling (OM). First, we look at the basic Q-learning setting without OM. Second, we implement the basic OM learning scheme called Fictitious Play. And lastly, we will attempt to improve general performance of the agents by combining Q-learning and Fictitious Play.

1.1 Q-learning

Without OM, each player only has knowledge of the three possible actions, Rock (R), Paper (P) or Scissors (S). After each episode, the players are given a reward which they use to update their Q-values and adapt their strategy. All possible agent strategies are pareto-optimal as RPS is a zero-sum game. This means that no agent can increase his result without negatively affecting another agent. The unique Nash-Equilibrium is the mixed-strategy $(33\%, 33\%, 33\%)$. In other words, there is no incentive for a player to deviate from playing at random. In doing so, his opponent could improve his total utility by countering the deviation, triggering our first player to immediately return to the equilibrium.
In our experiments, we have set up the agents with Boltzmann exploration, so the probability of playing action a is given by

$$P(a) = \frac{e^{Q(a)/T}}{\Sigma_{a' \epsilon A} e^{Q(a')/T}}. \tag{1}$$

We have chosen the temperature $T = 10/min(episode, 120)$. Limiting the temperature from dropping below 1/12 allows a stable yet very minimal degree of exploration after the onset of a game. Table 1 shows the probability of a strategy profile to be played on average after 10 games of 20000 episodes each. The *Total* column and row provide evidence that the Nash-equilibrium is indeed reached for our Q-learning players. There is an interesting observation to be made in how they have acquired this strategy. Both players inevitably seem to get stuck upon playing the same action. Due to the minimal exploration, they eventually escape from such a strategy profile until they once more end up playing the same action. We believe a possible explanation for this behaviour is that, after a losing streak on one action, the agent abandons this action and tries out one of the other actions. Eventually ending up with another losing streak here, the agent turns to another action. This pattern continues until both agents end up playing the same action, resulting in a stable reward of zero, which causes the dynamics of their Q-values to temporarily freeze, in turn causing static behaviour. It is clear that, without Opponent Modelling, the agents obtain no accurate perception of the environment which changes as a result of their own and another agent's actions. Figure 1 shows the accumulated reward of a single game played using Q-learning, and illustrates these discussed periods of static behaviour. Figure 2 visualises the dynamics trajectories of the RPS game.

	R	**P**	**S**	**Total**
R	27.63%	3.50%	3.53%	34.66%
P	3.43%	25.351%	3.43%	32.21%
S	3.43%	3.53%	26.16%	33.11%
Total	34.49%	32.39%	33.11%	

Table 1: Probability of strategy profiles after 10 games of 20000 episodes each using Q-learning (R = Rock, P = Paper, S = Scissors). The *Total* row and column show the mixed strategy used by respectively the column and row player.

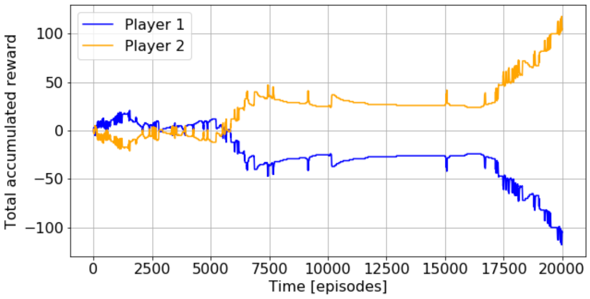


Figure 1: Total accumulated reward for two competing Q-learning agents

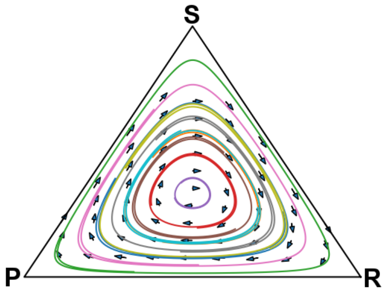


Figure 2: Evolutionary dynamics for the Rock-Paper-Scissors game

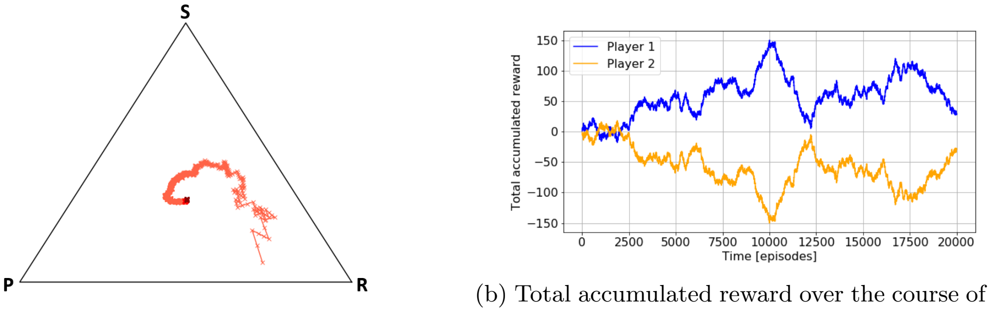
1.2 Fictitious play

In fictitious play (FP) [6], every player maintains a record, or belief state, based on the history of the opponent's moves, and uses this belief state to predict that the opponent will play action a next with probability

$$P(a) = \frac{w(a)}{\Sigma_{a' \epsilon A} w(a')} \quad (2)$$

where $w(a)$ is the number of times the opponent has played action a in the past. In other words, a player assumes the opponent to follow a stationary mixed-strategy. Based on these probabilities, a player can calculate the probabilities he should use for choosing his own actions in order to maximise his reward. Figure 3a shows the evolution of a belief state, initiated with a random belief. It is clear that FP finds its way to the Nash-equilibrium. This should come as no surprise, as explained in [6], *"Fictitious play was actually not proposed initially as a learning model at all, but rather as an iterative method for computing Nash equilibria in zero-sum games"*.

Figure 3b shows the total accumulated reward for our players during one game. In comparison to Figure 1, we observe the reward to follow a trajectory which is a lot less static or smooth, suggesting a higher degree of randomness in the strategy profiles played. Table 2 verifies this observation.



(a) History of belief about the opponent's strategy.

(b) Total accumulated reward over the course of a single game

Figure 3: Fictitious Play. In 3a, every red cross represents an update of the belief state. Darker red represents later points in time.

	R	**P**	**S**	**Total**
R	11.13%	11.08%	10.80%	33.00%
P	11.12%	11.04%	10.96%	33.12%
S	11.15%	11.46%	11.27%	33.88%
Total	33.40%	33.58%	33.02%	

Table 2: Probability of strategy profiles after 20000 episodes using Fictitious Play (R = Rock, P = Paper, S = Scissors). The *Total* row and column show the mixed strategy used by respectively the column and row player.

1.3 Opponent modelling

In this last section, we will combine Q-learning with the Fictitious Play learning rule. We call a player that combines these a QOM player or QOM agent (QOM = Q-learning + Opponent Modelling). Such an agent differs from our previous Q-learning agent, in that its Q-function now takes two arguments as input, respectively its own and its opponent's action (note that both Q-functions do not require a state argument as RPS is a single-state game). A QOM agent will use it's Q-function and Fictitious Play to calculate the Expected Value (EV) for playing action a as

$$EV[a] = \sum_{a^o \epsilon A^o} Q(a, a^o) * P_{A^o}[a^o] \tag{3}$$

where A^o is the opponent's actions space, a^o a single action within A^o and $P_{A^o}[a^o]$ the probability for the opponent to play action a^o. These probabilities are obtained with equation (2). Finally, applying the Expected Values instead of Q-values to the Boltzmann exploration equation (1), a QOM player determines the distribution underlying his next action.

When two QOM players are playing against one another, the emerging strategy profiles follow the same probability pattern as for two FP players (see Table 2). In other words, the Nash-Equilibrium is the preferred mixed-strategy for QOM players. What is perhaps more interesting is to observe the difference in how QOM players approach the equilibrium in comparison to FP players. To this end, Figure 4 displays the evolution of beliefs in a game where a QOM player faces a FP player. We clearly observe that it takes the QOM player longer to converge to the equilibrium point. To understand this behaviour, we have to remember that the FP player has intrinsic knowledge of the reward function, while the QOM player must learn the reward function first by means of Q-learning. We can therefore conclude that the QOM player inherits the best of both worlds. It is superior to Q-learning in that it does not display the same static behaviour as observed in section 1.1. It is superior to Fictitious Play in that it does not require any prior knowledge of the reward function.

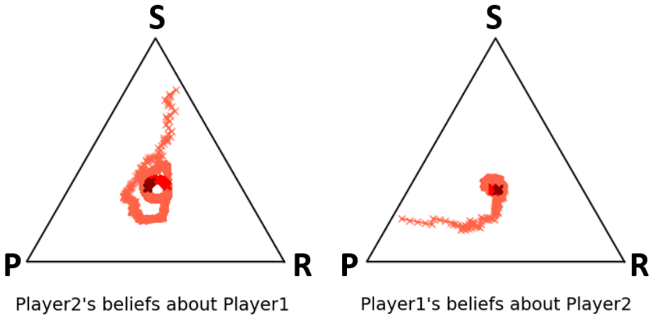


Figure 4: History of belief of the opponent's mixed strategy. Darker red represents later points in time. Player1 combines Q-learning with Fictitious Play (QOM player), Player2 is only equipped with Fictitious Play (FP player).

2 Task 3: Harvest Game Agent

2.1 Abstract

Commons dilemmas pose extraordinary challenges to reinforcement learning. As optimal behaviour in such situations cannot solely be based on short-term rewards, aspects of Game Theory have recently been brought up to aid agents in their quest for rationality in these complex multi-agent settings. In this research, we investigate the concept of Inequity Aversion. In a first setting, agent's possess inherent incentives for both competition and cooperation. We show that Inequity Aversion here allows agents to optimize their own reward while simultaneously abstaining from exhausting the common resource. In a second setting, we attempt to have agents reinforce the will of the group onto each other, providing the incentive for cooperation by means of punishment. We show that it is indeed possible for agents to learn sustainable behaviour in this setting under the condition of self-play.

2.2 Introduction

In the Harvest Game (or Apples Game [3]), a number of agents compete to collect apples. However, apples in the orchard only regrow as long as other apples are present. This leads the agents into a so called social dilemma. When agents deplete the shared resource of apples for personal benefit, they potentially act against the common good. In other words, a social dilemma is characterized by a tension between the interest of the group versus that of the agent.

A first and foremost question in such an environment is: if acting in self-interest is acting against the common good, what behaviour then is desired? For simplicity, We will assume here that the main goal of an individual agent is to collect as many apples as possible, and the main goal of the group as a whole is to maximize the total apple consumption. Therefore, agents who refuse to eat apples all together are as little as supportive towards the group as those whom harvest tirelessly. The desired agent behaviour is then a combination of both competition and cooperation. To this end, a technique known as Inequity Aversion has been applied to make agents aware of the group and their position within it. How can Inequity Aversion be modelled in the action-reward feedback circuit of an agent, and are Inequity Averse agents able to abstain from short-term personal gain in the face of a commons tragedy?

2.3 Approach

2.3.1 Deep Reinforcement Learning

The Harvest Game provides an environment which can essentially be formulated as a Markov Decision Process (MDP). In recent years, such MDPs have been the main playground for developing Reinforcement Learning (RL). However, for complex MPDs as the Harvest Game, the action-state space is far too big to maintain a Q-table as required in Q-learning. To be able to use Q-learning in such an environment, function approximation in the form of a neural network is used to provide an estimate of the Q-values corresponding to a given state.

In its most simple form, a Deep Q-network will be trained by calculating the target value Y_k^{DQN} for $Q(s, a; \theta_k)$ as

$$Y_k^{DQN} = r + \gamma * \max_{a' \epsilon A} Q(s', a'; \theta_k) \tag{4}$$

where s is the current state, a the action chosen, r the given reward and s' the next state reached. γ is the discount factor and θ_k refers to the parameters defining the Q-values at the k^{th} iteration.

As noted in [1], equation (4) does not guarantee convergence because updating the weights of the network also changes the target value. This is known as chasing a non-stationary target. To solve this issue, as well as handling the overestimation of the target value due to the max operator, a Double DQN (DDQN) network uses 2 networks: a primary with weights θ_k for selecting actions and a secondary (or target network) with weights θ_k^- to calculate the Q-values and update the primary during learning. θ_k^- is updated after a number of iterations to θ_k. This prevents divergence of the target values as these are kept fixed for a while. In mathematical form, the target value Y_k^{DDQN} is given by

$$Y_k^{DDQN} = r + \gamma * Q(s', arg\max_{a' \epsilon A} Q(s', a'; \theta_k); \theta_k^-). \tag{5}$$

The shape of our network for the first setting of the Harvest Game is shown in Figure 5. An input grid is created where known apples to the snake are represented by a positive '1'. The positions are always transformed in a way such that the snake is at the center of the grid facing forward/up. In the second setting of the Harvest Game, players are added to the state, represented by a negative '-1'. The network outputs the approximate Q-values for each possible action corresponding to the given state.

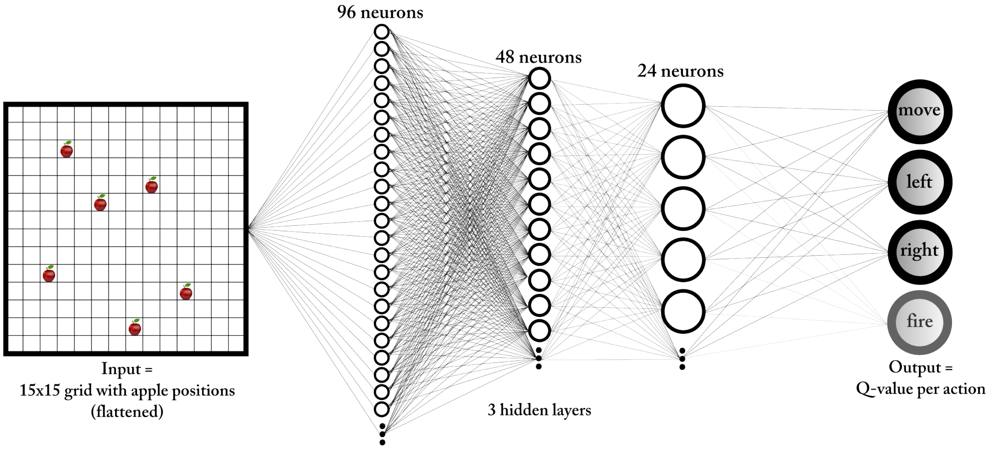


Figure 5: Neural network design

2.3.2 Inequity Aversion

In the first setting considered of the Harvest Game, we set the agents' action space A to $\{Move, Left, Right\}$, disallowing players to tag each other. Inequity Aversion must therefore be modelled explicitly into the agent, separately from the underlying DDQN network, effectively separating the agent's concern for competition and cooperation.

To achieve this behaviour, first a network is trained with the sole purpose of consuming apples. In a second stage, two types of agents are designed on top of this network who differ only in the way they translate Q-values to actions. Free-Riding (FR) agents have no incentive to deviate from choosing the action with the highest Q-value, therefore maximizing their short-term reward. Inequity Averse

(IA) agents however, are aware of the group and their position within it. Derived from the history of rewards of all N agents in the field, a player i is troubled by a sense of envy and guilt, defined after [2] as

$$envy_i^{t+1} = \frac{\alpha}{N-1} \sum_{j!=i} max(e_j^t(s_j^t, a_j^t) - e_i^t(s_i^t, a_i^t), 0) \tag{6}$$

$$guilt_i^{t+1} = \frac{\beta}{N-1} \sum_{j!=i} max(e_i^t(s_i^t, a_i^t) - e_j^t(s_j^t, a_j^t), 0) \tag{7}$$

$$e_j^t(s_j^t, a_j^t) = \gamma * e_j^{t-1}(s_j^{t-1}, a_j^{t-1}) + r_j^t(s_j^t, a_j^t) \tag{8}$$

where $e_j^t(s_j^t, a_j^t)$ is the temporal smoothed reward for agent j. The probability for an agent i to act envious at time t is now given by

$$P_i^t[envious] = \frac{envy_i^t}{envy_i^t + guilt_i^t} \tag{9}$$

and the probability to act guilty is then simply $P_i^t[guilty] = 1 - P_i^t[envious]$. Note here that the parameters α and β in equations (6) and (7) can be used to modify the relative weight of envy compared to guilt. Acting envious/guilty translates to playing the action with respectively the highest/lowest Q-value.
To summarize the first setting: Q-values are designed to reflect the agent's impulse towards short term gratification. An IA agent will reflect rather rationally on its emotional impulse, weighing envy and guilt to guide its course of action.

Second setting In the second setting of the Harvest Game, we abandon all inequity aversion modifications. The main purpose is to allow agents to tag each other, and to investigate if agents are able to maintain a healthy apple growth rate on their own. As tagging causing a minor decrease of 1 point to the tagger, and a major decrease of 50 points to the tagged, the rationale here is that tagging may spread confusion in the network's tendency to chase apples.

2.4 Results

2.4.1 Metrics

Next to **mean apple consumption** and **time steps until game-over**, we use a number of alternative metrics to evaluate the performance of the agents.

Inequality The Gini Coefficient provides an accurate representation (on a scale from 0 to 1) of the inequality between the final scores of different agents. Scores are influenced by the consumption of apples as well as tagging. For N agents and final score R:

$$Inequality = \frac{\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} |R_i - R_j|}{2N \sum\limits_{i=1}^{N} R_i} \tag{10}$$

Sustainability Sustainability is defined after [2] as the average time step at which rewards are collected, showing whether or not agents try to maintain a healthy apple concentration. If the sustainability is below 500, more apples were eaten in the beginning of the game then at the end, indicating an imbalance between individual profit and group interests.

Aggressiveness The average number of fire actions per agent.

2.4.2 Harvest Game: the first setting

In the first setting of the Harvest Game, agents are not allowed to tag each other and IA agents use envy and guilt explicitly to determine the underlying probability of their actions. In our experiments, we deploy 5 agents in the orchard, which initially contains 50 apples. The apple growth rate is kept fairly low (only an empty position with 1 neighbouring apple has a 0.8% change of growing an apple, positions with 0 or more than 1 neighbouring apple have a 0% growth rate). This setting removes the ”point of no-return” where apples become an inexhaustible resource.

Figures 6 and 7 show the results of our experiments to investigate the influence of Inequity Aversion in the presence of a social dilemma. The horizontal axis is defined as the proportion of envy within the inequity averse strategy. Therefore, FR agents in fact represent the special case where $\alpha/(\alpha+\beta) = 1$. The given plots are obtained by selecting 13 possible α/β combinations and averaging the given metrics for each combination over 20 games.

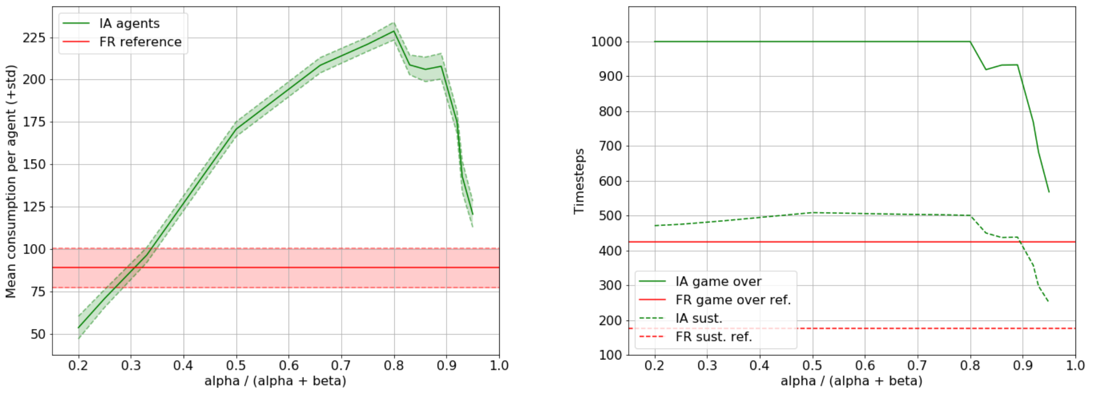
Figure 6a shows a strong upward trend of the mean consumption up until envy's share ticks off 80%. This is what we will call the breaking point for Inequity Aversion in this setting of the Harvest Game. The steep decline that follows becomes more meaningful if we consider Figure 6b. Here we see that beyond the breaking point, IA agents begin to overgraze the orchard, prematurely ending the game. The breakdown point can be predicted by the green-dotted Sustainability line. After Sustainability reaches it highest point at an equal proportion of envy and guilt, it slowly but steadily decreases. At the breaking point, the expected time for average consumption crosses the game's midway point of 500 time steps in a downward motion, suggesting too much consumption is bound to happen too soon. The destiny for ever increasing envious agents is shown by the FR reference. Here too, the sustainability line is below the midway game-over time, suggesting the orchard's depletion at a later stage. Lastly, Figure 7 displays the average inequality in the group of agents at the end of the game. Interestingly, inequality is at its lowest when agents are proportionally more envious than guilty, appealing to the doctrine that individual ambition serves the common good. Video reference is provided for both FR and IA agent behaviour in this setting of the Harvest Game in [4] and [5].

2.4.3 Harvest Game: the second setting

Results In the second setting of the Harvest Game, the tag action is enabled and 10 agents are set out in the field. The discount rate (gamma) is set to 0.99 to ensure a long-term decision-making process of the agents. Our main goal is to investigate the learning process of our model and the evolution of the agent's behaviour. The results of our experiment is shown in Figures 8a, 8b and 9.

First exposure and early stage model It is clear that the agents learn to focus on the apples after only a few iterations. The fire plots show that the agents find no use in the fire action. However, as shown in Figure 8a, the lack of apples in the late game due to fast consumption often results in a premature end of the game. This observation can also be supported by the sustainability plot, where we see that most of the apples are consumed in the beginning, versus almost none at the end. It is important to note here that the scores are significantly lower than in the first setting. This is due to the fact that we introduced the fire action. Even if only a few agents are hit every episode, this results in a huge diminution of the total and average score.

Late stage model After a while, we start to see that the model begins to react to the scarcity of apples, and it slowly converges to a more sustainable strategy. This is purely done by avoiding



(a) Mean apple consumption per agent (b) Sustainability and time steps until game over

Figure 6: Influence of the hyper-parameters α and β on mean apple consumption, time steps until game over and corresponding Sustainability (first setting of the Harvest Game).

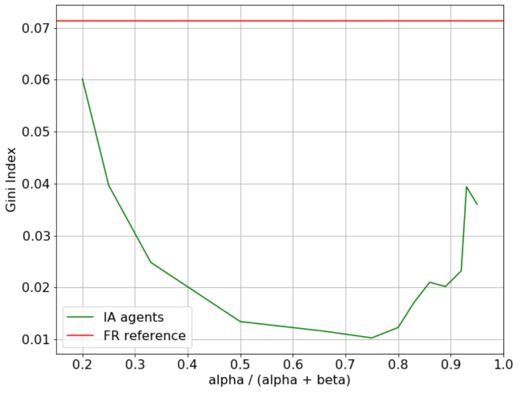
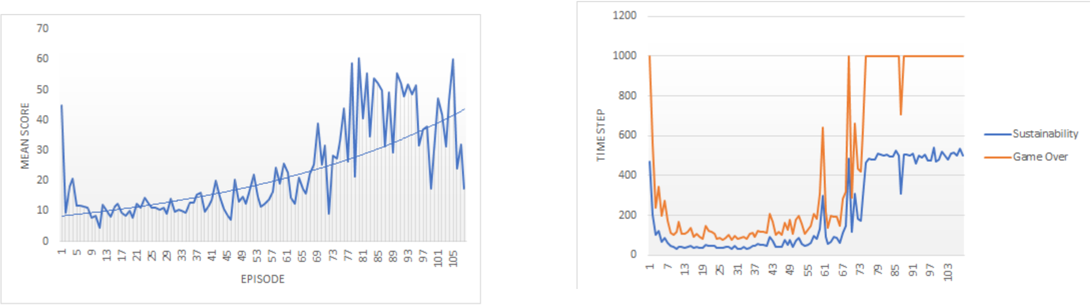


Figure 7: Inequality between individual agents measured by the Gini index (first setting of the Harvest Game).

apples from time to time, and the fire action seems to regain some use (see aggressiveness plot). The sustainability plot shows that as many apples are consumed in the first half of the game as in the second half, indicating a stable apple consumption.

Conclusion and remarks Our network is certainly able to adapt itself to handle the apple scarcity and maintain a healthy rate at which apples are consumed. The fire action doesn't bring much advantage to anybody, and is therefore mostly forgotten. An important aspect to keep in mind here is that all agents in our setting used the same algorithm, so they were sure that everyone would follow the same policy. If one of the agents follows another policy and depletes the apples, our



(a) Mean Score of agents

(b) Plot showing sustainability and time step at which game ends.

Figure 8: Different evaluation metrics: mean score (a) and sustainability (b)

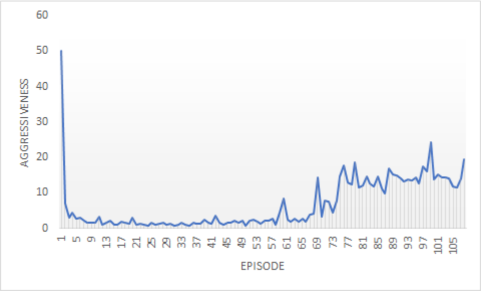


Figure 9: Plot showing aggressiveness of agents

model won't be able to react accordingly. However, as long as everyone is guaranteed to cooperate, our model seems able to learn a healthy strategy on its own.

References

[1] Vincent François-Lavet et al. "An Introduction to Deep Reinforcement Learning". In: *CoRR* abs/1811.12560 (2018). arXiv: 1811.12560. URL: http://arxiv.org/abs/1811.12560.

[2] Edward Hughes et al. "Inequity aversion improves cooperation in intertemporal social dilemmas." In: *CoRR* abs/1803.08884 (2018). arXiv: 1803.08884. URL: http://arxiv.org/abs/1803.08884.

[3] Pieter Robberechts et al. *The Apples Game*. Accessed: 2019-05-11. URL: https://github.com/ML-KULeuven/the_apples_game.

[4] Hendrik Serruys. Accessed: 2019-05-13. URL: https://youtu.be/lSGcOi3GMxY.

[5] Hendrik Serruys. Accessed: 2019-05-13. URL: https://youtu.be/9IZUUiXm1hU.

[6] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521899435.