Multi Agent Reinforcement Learning for Cooperative Hunting Scenarios

Sudeep Katakol
BITS-Pilani, Goa, India
f20150001@goa.bits-pilani.ac.in

Shikhar Rastogi
BITS-Pilani, Goa, India
f20150532@goa.bits-pilani.ac.in

Saloni Dash
BITS-Pilani, Goa, India
f20150292@goa.bits-pilani.ac.in

Megha Dhanuka
BITS-Pilani, Goa, India
f20150589@goa.bits-pilani.ac.in

Rishi Raj, Grandhe
BITS-Pilani, Goa, India
f20150544@goa.bits-pilani.ac.in

***Abstract*—This paper aimed to bring out the evolution of cooperation in our society by examining the cooperative hunting scenarios in lions extensively studied by biologists over the years. The objective was to simulate the random behaviour of animals and show how their actions converge/diverge under different conditions. The simulation was done using Multi Agent Reinforcement Learning. The agents learnt their rewards through Nash Q-Learning, and we were able to simulate three different scenarios under different conditions - one in which the animals fight, one in which they cooperate, and one in which they mix both these strategies.**

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

I. Introduction

Our work revolved around the scenario where two lions exist in a limited area, and can choose either to cooperate to catch the prey and share the reward, or kill off/injure the opponent and then keep all prey for themselves. The prey is assumed to be part of the environment and the agents are the players.

To simulate this game, our first job was to design a theoretical model based on real world data, on which we would run our computational MARL model. To do this, we first analyzed similar papers in the past, which had talked about the cooperative hunting scenario. In the paper, Evolution of Cooperative Hunting, Packer and Ruttan examined several Evolutionary Stable Strategies for the agents (hunters, in this case). Their paper considered the following possibile roles i.e. actions each agent could have/play -

- Cooperator
- Cheater
- Scavenger
- Solitary

They came up with their own payoff matrices, with parameters such as prey encounter rates, prey capture rates, costs of pursuing and costs of subduing prey. Taking inspiration from these, we came up with two models.

II. Our Contributions

Our first model uses payoffs from the matrix given by Packer and Ruttan, with some modifications. However, instead of a solitary agent, we have come up with a new role in which the agent has to first eliminate(fight) its opponent to play solitary. So the agent has to incur a cost to be able to hunt alone.

P1/P2	**Cooperate**	**Fight**
Cooperate	L$_2[H_2(V/2-C_2)-E_2]$	0, L$_1[H_1(V-C_1)-E_1]-K$
Fight	L$_1[H_1(V-C_1)-E_1]-K, 0$	L$_w[H_w(V/2-C_w)-E_w]$

TABLE I: Payoff Matrix for Cooperate Scenario

(Cooperate,Cooperate) and (Fight,Fight) have the same utilites for both agents, and are hence denoted by single values. In this payoff matrix, subscript one denotes the value for a solitary agent, where as subscript two denotes the value for the pair of agents.

(Fight,fight) has subscript w which denotes extremely low valued real numbers, and represent the case when both predators decide to fight each other and end up wounding each other. This means that the probability they catch a prey is extremely low now.
L_i = Prey encounter rate
H_i = Probability of catching prey
V = Value attached to prey in one particular hunt
C_i = Cost of subduing prey
E_i = Cost of pursuing prey
K = Cost incurred to kill off the other agent

We assume that the predators split the reward(V) equally when they cooperate. Also, in this payoff matrix all values and probabilities are actually functions of the number of prey(n) available. For example, when n is very small, the value attached V to a single hunt will be extremely high as both predators value the hunt and its reward a lot and it is crucial for them to succeed in the hunt. Conversely, when n is very high, the V will be lower, as the predators do not attach a lot of value to a particular hunt as they feel they have more chances at catching prey.

Now let us look at further modifications we made to this particular model which was heavily inspired from the Packer and Ruttan model of Cooperative Hunting.

III. Modified Game

The Second model which we have designed to show the Hunting Scenario is based on the similar principles discussed above. It has two different cases depending on how powerful each predator is. In the first case both the predators are considered as equally powerful and the cost to catch a prey is equal. In the second case one predator is more powerful than the other and thus the payoff matrix changes accordingly.

A. Case 1

P1/P2	**Cooperate**	**Fight**
Cooperate	(K-C)N/2, (K-C)N/2	0, (K-C)/N
Fight	(K-C)/N, 0	(K-C-I)N/2, (K-C-I)N/2

TABLE II: Payoff Matrix for Case 1

N = Total number of prey
K = Value assigned to each prey
C = Cost incurred to kill a prey
I = Loss of energy during fight

As mentioned before both the predators are equally powerful and thus they equally divide the payoff in case of (Cooperate,Cooperate) and (Fight,Fight). When either of the one tries to Cooperate but the other one wants to fight or vice-versa, the value of payoff for the one who wants to Cooperate is zero, as he did not get cooperation from the other predator to catch the prey.

B. Case 1

P1/P2	**C**	**F**
C	(K-C$_1$)$NC_2/(C)$, (K-C$_2$)$NC_1/(C)$	0, (K-C$_2$)$/N$
F	(K-C$_1$)$/N, 0$	(K-C$_1 - I_1)NC_2/(C_2)$ (K-C$_2 - I_2)NC_1/(C)$@c@

TABLE III: Payoff Matrix for Case 2

N = Total number of prey
K = Value assigned to each prey
C_i = Cost to catch a prey for the ith player
C = C$_1 + C_2$
I$_i$= Loss of energy for the ith player during fight
I$_1 = I(C_2)/(C_1 + C_2)$
$I_2 = I(C_2)/(C_1 + C_2)$

In this second case, if we assume P1 is more powerful than P2, then C1 will be less than C2, as he will catch the prey more easily. The total value during (Cooperation,Cooperation) is not equally divided among the players but in proportional to their capability. The case of (Fight,Cooperate) and (Cooperate,Fight) is similar to Case 1, with different values of C1 and C2.
The case of (Fight,Fight) is quite interesting as the loss of energy during fight is not equal for P1 and P2, but in fact depends on how vigorous they are.

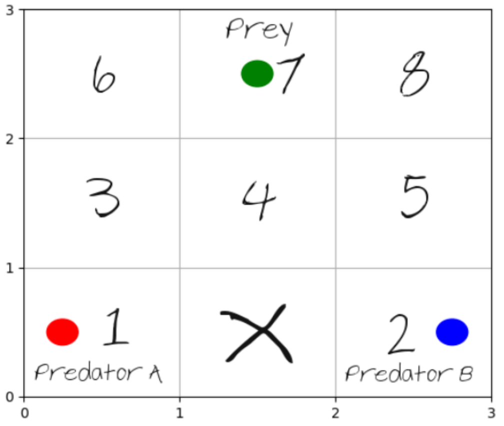


Fig. 1: Environment

IV. Modeling

We modeled the the scenario into a grid based environment as shown in Fig 1. Predator A and B start at positions 1 and 2 respectively and they both have to catch the prey, which is at position 7. Prey is blissfully unaware of the existence of the predators. For simplicity, we have assumed that the predator A can move only right and up while predator B can move left and up. If predator A decides to move left from position 1 he ends up in position 4. The same goes for predator B if he decides to move right from position 2.

The goal of the predators is to catch the prey. However, the game also ends when both of them decide to fight each other out and end up in position 4 after the first move. The rewards are given to both the predators only when the game finishes depending on how they hunt the prey.

V. Methods

To train our predators to choose the best possible option available to them we use a technique called as Nash Q learning. Nash Q learning is a Reinforcement Learning algorithm and is a modification to the Q learning update rule which allows it to be extended when dealing with multiple agents in reinforcement learning.

A. Reinforcement Learning

In Reinforcement learning, an agent tries to learn the optimal control policy by interacting with the external world. The external world is modeled as a discrete-time, finite state, Markov decision process (MDP). Each state, action pair (s, a) is associated with a reward. At each time step, the agent observes a state s, chooses an action a, receives a reward

$r(s,a)$, and transitions to a new state $s^{'}$. The task of reinforcement learning is to maximize the long-term discounted reward per action, by balancing exploitation (always choosing those actions that give it the highest reward) and exploration (try new actions that might give better returns in the long run)

B. Q Learning

Q-Learning is an approach to incrementally estimate the utility values of executing an action from a given state by continuously updating the Q-values using the following equation:

$$Q(s,a) = Q(s,a) + \alpha(r(s,a) + \gamma(\max_{a'} Q(s',a') - Q(s,a))) \tag{1}$$

$$Q(s,a) = (1-\alpha)\ Q(s,a) + \alpha\ (r(s,a) + \gamma\ \max_{a'} Q(s',a')) \tag{2}$$

where $Q(s,a)$ denotes the utility of taking action a from state s,γ is the discount factor and α is the learning rate.

Watkins and Dayan(1992) have shown that Q-learning algorithm converges to an optimal decision policy for a finite Markov decision process.

C. Nash Q Learning

Nash Q-learning is an extension of Q learning to the multi agent reinforcement scenario. In Multi Agent reinforcement learning (MARL), the external world seen by an agent can't be modelled by an MDP and is instead modelled as a Stochastic Game. Every agent's value function at a state s depends on the actions taken by all the other agents too. The next state, s' which each of the agent reaches is also the culmination of all the agents' actions. Hence, the Q values of every agent form a game whose payoffs become the value function of the corresponding agent. Nash Q learning assumes that the optimal policy at each step is the Nash Equilibrium for the above game. And thus the Nash Q learning update rule reduces to the following for two agents:

$$\begin{split} Q(s,(a,b)) = (1-\alpha)\ Q(s,(a,b)) + \alpha\ (r(s,(a,b)) + \\ \gamma\ NashEq_{a',b'} Q(s',(a',b'))) \end{split} \tag{3}$$

VI. Experimental Setup

The environment for the agents' to train in was built from scratch, keeping in mind OpenAI Gym standards, for seamless integration with other powerful Reinforcement Learning libraries like Keras-Rl.

OpenAI Gym[1] is an open source library for developing Reinforcement Learning Algorithms and giving the agents a platform to learn by interacting with the environment.

As mentioned previously, the Nash Q Learning algorithm was used to train the agents. The Algorithm was implemented in scratch in the programming language of Python[2] The hyperparameters used for training the agents with the algorithm are:

- alpha (α) = 0.3
- discount factor (γ) = 1.0
- number of training episodes = 10000
- exploration rate = 1.0
- exploration rate decay = 0.001

VII. Results

We tested three scenarios, the first one in which the Nash Equilibrium was Cooperation, the second one in which Fight was the Nash Equilibrium and the final one in which there were multiple Nash Equilibria.

A. Cooperate

The payoff matrix for the first scenario is given in the table below. The Nash Equilibrium here is (Cooperate, Cooperate), and we do in fact observe our agents converging to that behaviour.

P1/P2	**Cooperate**	**Fight**
Cooperate	3, 3	0, 2
Fight	2, 0	-1, -1

TABLE IV: Payoff Matrix for Cooperate Scenario

B. Fight

The payoff matrix for the second scenario is given in the table below. The Nash Equilibrium here is (Fight, Fight). We observe that our agents' behaviour does converge to the Nash Equilibrium.

P1/P2	**Cooperate**	**Fight**
Cooperate	3, 3	0, 4
Fight	4, 0	1, 1

TABLE V: Payoff Matrix for Fight Scenario

C. Multiple Nash Equilibrium

The payoff matrix for the final scenario is given in the table below. In this scenario, the Nash Equilibria are (Fight, Fight) as well as (Cooperate, Cooperate). The agents randomly choose to fight or to cooperate. The converged behaviour alternates between Fig 2, and Fig 3 and other ways to cooperate. [3].

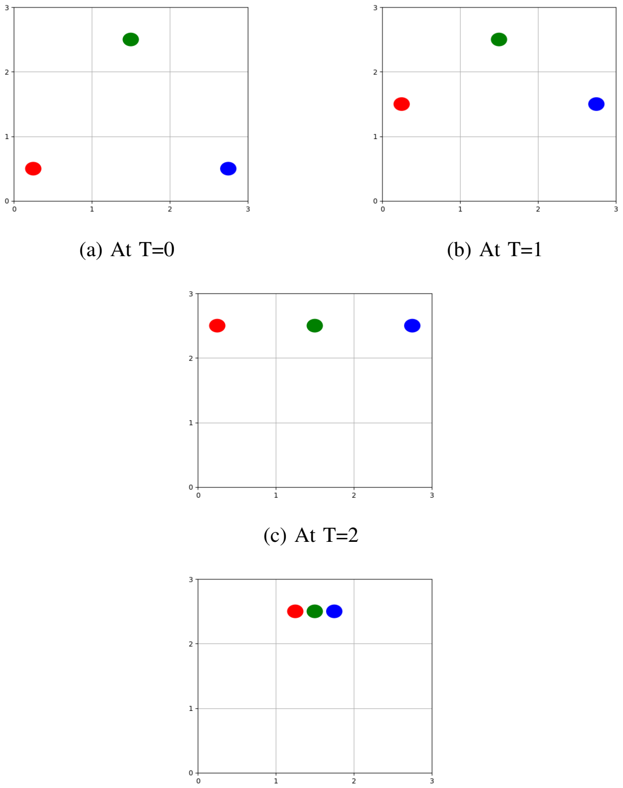
P1/P2	**Cooperate**	**Fight**
Cooperate	3, 3	0, 2
Fight	2, 0	1, 1

TABLE VI: Payoff Matrix for Multiple Equilibria Scenario

[1] More Information can be found here : https://gym.openai.com/

[2] The complete code can be found here: https://github.com/sudeepkatakol/MARL-CooperativeHunting

[3] Visit the github link : https://github.com/sudeepkatakol/MARL-CooperativeHunting for more figures



(a) At T=0

(b) At T=1

(c) At T=2

(d) At T=3

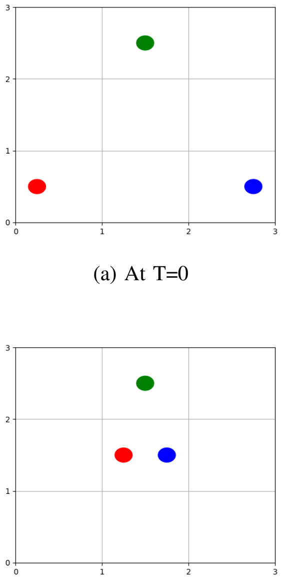
Fig. 2: Converged behavior of the two predators as Cooperation

VIII. Real World Applications

While this paper uses cooperative hunting to devise a model for predators and prey, it should be noted that the underlying premise on which this is based on is that of Supply and Demand. The context of this paper can indeed be extrapolated to any situation or scenario where there exists a scarcity of resources and more than one competitor.

A very direct context that can be derived from our paper is that of economic trade. More specifically, a Duopoly model. Similar to our model, there exist two firms who are competitors for the same market and can either choose to cooperate or try to exploit the market on their own. The results obtained from our experiments can indeed be observed in this particular context. In situations of cooperation, we do observe firms working together on fixing a certain price level or a certain supply level with a promise to not undercut one another and in situations of Fighting, we do observe situations escalating to price wars.

Another interesting application of our methods and model would be in the field of Anthropology and Evolution. There is significant evidence that humans and other species of animals pass down information through their genes enabling them to learn certain behavioural patters that enable them to live better and longer. The methods used in this paper can be utilized to model such behaviour and in some cases even predict future behaviour to aid better understanding of the workings of evolution.



(a) At T=0

(b) At T=1

Fig. 3: Converged behavior of the two predators as Fight

IX. Future Work and Conclusion

The implications of the methods and ideas explored in this paper do have far reaching consequences. A direct increase in complexity can be achieved by simply increasing the number of predators and prey in our model. The problem with solving large scale game theory problems for the Nash Equilibrium with traditional techniques is that the mathematical complexity falls under the category of PPAD which lies along the line of np-hard problems. The techniques used in this paper however can be used to reach the Nash Equilibrium in a much shorter duration.

By utilizing the power of RNNs, we can solve similar large scale complex problems in the fields of economics and businesses. Future work on this paper would include making models with multiple predators and prey with additional constraints on movement and/or actions, modifying the payoff matrix with additional variables, including actions for prey, etc.

Acknowledgment

We would like to thank Dr. Himadri Mukherjee for giving us the opportunity to work on such an interesting topic in Game Theory. Also, we would like to thank him for his excellent insights and guidance through out the semester because of which we were able to successfully complete our project.

We would also like to extend our thanks to our colleagues who assisted us during our work.

References

Packer, C. and Ruttan, L. (1988). The Evolution of Cooperative Hunting. The American Naturalist, 132(2), pp.159-198.

Hu, J. and Wellman, M. (2003). Nash Q-Learning for General-Sum Stochastic Games. Journal of Machine Learning Research. [online] Jmlr.org. Available at: http://www.jmlr.org/papers/v4/hu03a.html.

Watkins, C. J. C. H. (1989). Learning With Delayed Rewards. Ph.D. thesis, Cambridge University Psychology Department.

Watkins, C. J. C. H. Dayan, P. (1992) Technical Note: Q-Learning. Machine Learning, 8(3/4), Kluwer Academic Publishers.

Constantinos D., Paul W. and Christos H. (2008) The Complexity of Computing a Nash Equilibrium