Learning dynamics
Rational and Convergent Learning in Stochastic Games

Wassim Belgada, Iliass El Achouchi, Ibrahim Imestir and Aymane Mountasser

Abstract

This paper deals with the problem of policy learning of several agents in a stochastic environment. This problem can be addressed by reinforcement learning algorithms that are implemented according to two properties, namely: rationality and convergence. There is also a new rational reinforcement learning algorithm based on the principle of : "Learn quickly while losing, slowly while winning". We will discuss these two properties, the reinforcement learning algorithms and this new rational algorithm in this paper.

Introduction

The problem of multi-agent policy learning lies in the fact that some agents learn in an unpredictable way, making it difficult to design a reliable learning algorithm. Indeed, knowing that the agents will adapt to these agents whose learning is out of control, we risk to tend towards an inappropriate policy system if the learning algorithm used does not address this problem. Stochastic games were introduced in 1953 and have been the subject of several experiments in the field of game theory. Like most games where learning is possible, stochastic games are based on Markov decision processes whose appropriate level of control has been discovered through reinforcement learning. We will go through the rational algorithm evoked in the previous point allowing to face these problems in a framework of stochastic multi-agent games. We will then present the essential components of game theory (Nash Equilibrium, Good Answers, Stochastic Games), the two properties allowing the promotion of agent learning in a system, the new rational algorithm named "Win Or Learn Fast" (WoLF) as well as the results of reinforcement algorithms compared to our new rational algorithm.[(Michael. and Manuela)]

Methods

In this section, we will define our working environment, i.e. the algorithms we are going to use, such as WoLF-PHC, PHC etc..., but also what we are going to use them on, the stochastic games.

Stochastic games

As mentioned in the introduction section, a stochastic game is an extension of Markov decision processes. We add in our analysis the fact that the system is multi-agent, that their transition function to the next state is done via the meeting of all the agents and that each agent has its own reward function. The objective of each agent is to maximize its reward with a factor γ by selecting the right answers. A stochastic game is characterized by a tupe $(n, S, A_{1...n}, T, R_{1...n})$ where n is the number of agents, S is the set of states in the game, Ai is the set of actions available to the agents (A being the number of actions available to the system game), T is the state transition function, and Ri is the reward function of an agent. A matrix game is a game between a "row" player and a "column" player in a reward matrix (i.e. Matching Pennies and Rock-Paper-Scissors), each getting the indicated reward for a win or the indicated loss for a loss. We can posit that each state of a stochastic game can be represented as a matrix of rewards where the payoffs are determined by the matrix Ri present in the tuple, at the end of a "run" (change of state), the transition function changes the state of each agent of the system. A stochastic game thus has the characteristics of a matrix game and of Markov decision processes.

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$
Matching Pennies

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$
Rock-Paper-Scissors

Figure 1: Two example matrix games

Mixed Policies. Unlike single-agent situations, deterministic strategies in multi-agent situations can often be used by other agents. In the case of penny matching, if Player A played deterministically, Player B could always find a way to have a positive payoff. That's the reason why mixed policies should be considered, which take into account the probabilities on the actions of the other players. A mixed policy is a function of the form: $\rho : S- > PD(A_i)$.

Nash Equilibria. Having now the knowledge on the concept of mixed policies, we still do not know how to define an optimal strategy independent of the adversary strategies. And so we will define the concept of best response.A strategy is optimal regardless of the strategy used by the other player(s) if it is considered to be a better response against the other players' strategies . And so we can define a major advance in game theory, which is the Nash equilibrium.

A Nash equilibrium is the set of strategies for a player that are considered as the best response for the others player's startegies. Thus, no one can have a better payoff by changing their strategy.Knowing that all matrix games have such an equilibrium it becomes interesting to try to get as close to it as possible, and may even have several equilibria. In the games like the ones in figure 1, we can see that they have equilibria that consist in the fact that the player plays all the actions with the same probability for each of them.

Motivation

One of the downsides of multi-agent systems is that it is akin to a "moving target" problem. If the players who are not under our control change the way they play, this will also affect the behavior of our agent. Equilibrium solutions do not solve this problem because the agent does not know the equilibrium that the other players will play or that they have a bias towards equilibrium.

Moreover, developing a learning algorithm for our agents is difficult as we do not recognize what learning algorithms other learning agents are using. It is neither sensible nor practical to imagine a general case in which other players can completely change their policies. On the other hand, it is unacceptable to make restrictive assumptions about other players' specific modes of adjustment, as other learners are beyond our control and therefore we do not know what constraints to assume. We address this multi-agent learning problem by defining two characteristics of the learner that demand on his behavior in practical situations. First, we will introduce these features. Then, we study previous multi-agent reinforcement learning techniques and demonstrate that they do not meet these properties all at once.

Properties

We provide two useful properties of multiagent learning algorithms: rationality and convergence

Property 1 *(Rationality)* On condition that the other players' policies converge to stationary policies then the learning algorithm will converge to a policy that is more responsive to their policies. This is a very straightforward property that requires the player to perform best when other players are using stationary techniques. Players need to learn good responsive policies when they exist. Non rational algorithms prefer to learn certain policies independently of the policies of other players, for example, their part of some equilibrium solution. This does not work in multiple equilibria games, where agents cannot independently select and play an equilibrium.

Property 2 *(Convergence)* The learner will mandatory converge towards a stable policy. This property would normally be conditional on other agents using an algorithm of some kind of learning algorithm.

The second property,requisite that against some class of other player's learning algorithms, the learner's policy will converge. For instance, one can refer to convergence with respect to players with stable policies or convergence with respect to rational players.

In this article, we concentrate on convergence on the assumption of the self-play. That is, in the case all players use the same learning Turing machines; the question is therefore whether player policies converge. This is an important and not easy step towards convergence against more general classes of players. Furthermore, neglecting the eventuality of self-play leads to the simple assumption that other players are inferior as they cannot use a similar algorithm. In combination, these two these two characteristics guarantee that the learner will converge to a strategy that is stable but also optimal against the strategies of his opponents. There is also a relation between these characteristics and the Nash equilibrium.When all the players are rational, if they converge, they will have converged in a Nash equilibrium. Since all actors converge on a stable policy, each actor, being rational, must converge to obtain the best response to their policies. Since this is true for every player, then by definition their policies must be a equilibrium. Also, if all players are rational and convergent with respect to the other players' algorithms then they converge to Nash equilibrium.

Other Reinforcement Learners

There are some RL techniques that are directly related to learning in a multi-agent system. We look at three RL techniques: Single Agent Learners, Joint Action Learners (JAL), and Minimax-Q.

Single agent learner. Although it is not really a multi-agent learning algorithm, one of the more common ways is to implement a single-agent learning algorithm(e.g. Q-learning, TD(λ), prioritized sweeping, etc.) for a multi-agent domain.

They, of course, ignore the existence of other agents, believing that their rewards and changes are Markovian. They primarily consider other agents to be part of the environment. This simple method meets a feature. In case the

other agents play, or converge to, stable strategies then their Markovian assumption is true and they converge to an optimal response. Otherwise, it is not generally convergent in self-play.

This is evident to see for Turing machines that learn only deterministic policies. As they are rational, in case they converge. It have to be to a Nash equilibrium. If we are studying games in wich all the equilibria are actually mixed equilibria (e.g. Matching Pennies), they could diverge. There are single-agent learning algorithms able of playing stochastic policies. However, in general jif the learner has only the ability to play stochastics policies it will not be sufficient for convergence, as will be shown in Section 4.

The learners of joint action. JAL. JALs estimate the adversaries actions and make their choices taking this into account. They cannot converge to a mixed equilibria, but are rationals.

Minimax-Q. These algorithms focus on both the actions and rewards of other players and try to learn the Nash equilibrium clearly. Agents using Minimax-Q learn and play equilibrium independently of the actions of other players.

Taking the case of the Rock-Paper-Scissors game, if the agent plays against an opponent who always plays "Rock", our agent will converge to a stationary equilibrium, but it is not optimal because the opponent's strategy must be taken into account.

We will try to find a rational algorithm that converges to an equilibrium.

A New Algorithm

In this section we will introduce an algorithm aiming at having a convergent and rational agent. But first, we will show a rational algorithm capable of playing mixed strategies, but which does not converge. We will then modify it in order to have a rational algorithm that converges in our experiments.

Policy Hill Climbing Q-learning can be extended to hill-climbing policy, an algorithm that can play in mixed strategies. The algorithm is shown in figure 2. The algorithm, itself, applies hill-climbing in mixed strategies. The Q-Values are maintained as a simple Q-Learning. In addition, the algorithm maintains the current mixed strategy. The strategy is improved by increasing the probability that it selects the largest action according to a learning rate $\delta \in (0, 1]$. If $\delta = 1$, the algorithm is equal to the Q-Learning. This technique, like Q-Learning, is rational and will converge to an optimal strategy if the other players play stationary strategies. Because, like Q-Learning, Q values will converge to Q* with an appropriate exploration strategy. Same for π. So, PHC is rational and can play mixed policies, but it still does not guarantee convergence and we will see this later.

WoLF Policy Hill-Climbing Now we will see the main element of this article. The element follows two important

1. Let α and δ be learning rates. Initialize,
$$Q(s,a) \leftarrow 0, \qquad \pi(s,a) \leftarrow \frac{1}{|\mathcal{A}_i|}.$$
2. Repeat,
   (a) From state s select action a with probability $\pi(s,a)$ with some exploration.
   (b) Observing reward r and next state s',
   $$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a')\right).$$
   (c) Update $\pi(s,a)$ and constrain it to a legal probability distribution,
   $$\pi(s,a) \leftarrow \pi(s,a) + \begin{cases} \delta & \text{if } a = \text{argmax}_{a'} Q(s,a') \\ \frac{-\delta}{|A_i|-1} & \text{otherwise} \end{cases}.$$

Figure 2: Policy hill-climbing algorithm (PHC) for player i

principles: using a variable learning rate, and the WoLF principle. These modifications will be directly applied to the naive hill climbing strategy.

The idea is that convergence is encourage by varying the learning rate used by the algorithm, without neglecting rationality. The WoLF principle is an appropriate method. This principle follows this intuition, learn quickly when you lose, and learn more slowly when you win. The way to know when we are winning is to compare the current payoff of the strategy with that of the average policy over time. This principle helps to converge because during a victory, the opponent adapts, and our agent takes this into account, which slows down his learning, but he will modify his strategy in a stronger way during a defeat.

The changes made are in the algorithm contained in the figure 3. In practice, the algorithm requires two different learning rates. The learning rate used to update the Q-table depends on whether the agent is currently winning (δ_w) or losing (δ_t). This is indicated by comparing the expected value, using the current Q-value estimates, of following the current strategy π in the current state with the average strategy π.In the smaller case, we are currently losing, and thus use δ_t, the larger learning rate.

Moreover, we had to implement ourselves the contrain to a legal probability distribution indicated in the algorithm. This part being absent in the algorithms presented in the initial article, we did our best to have a modification as close as possible to what is expected.

The learning speed being the only thing changed, WoLF policy hill-climbing remains a rational algorithm. These are not changes that allow to remove the rationality. The convergence property however, is quite different. The next section will show results on several stochastic games and will thus indicate the rationality of our method. Wolf has been stud-

1. Let $\alpha, \delta_l > \delta_w$ be learning rates. Initialize,

$$Q(s,a) \leftarrow 0, \qquad \pi(s,a) \leftarrow \frac{1}{|\mathcal{A}_i|}, \qquad C(s) \leftarrow 0.$$

2. Repeat,

(a,b) Same as PHC in Table 1

(c) Update estimate of average policy, $\bar{\pi}$,

$$\begin{aligned} C(s) &\leftarrow C(s)+1 \\ \forall a' \in \mathcal{A}_i \quad \bar{\pi}(s,a') &\leftarrow \bar{\pi}(s,a') + \\ &\quad \tfrac{1}{C(s)}\left(\pi(s,a') - \bar{\pi}(s,a')\right). \end{aligned}$$

(d) Update $\pi(s,a)$ and constrain it to a legal probability distribution,

$$\pi(s,a) \leftarrow \pi(s,a) + \begin{cases} \delta & \text{if } a = \text{argmax}_{a'} Q(s,a') \\ \frac{-\delta}{|A_i|-1} & \text{otherwise} \end{cases},$$

where,

$$\delta = \begin{cases} \delta_w & \text{if } \sum_a \pi(s,a)Q(s,a) > \sum_a \bar{\pi}(s,a)Q(s,a) \\ \delta_l & \text{otherwise} \end{cases}.$$

Figure 3: WoLF policy hill-climbing algorithm for player i

ied and is restricted for some types of games. Two player games, two action games, iterated matrix games, gradient ascent (which are known to be non-convergent). All these types of games are guaranteed to converge to a Nash equilibrium using a variable learning rate according to the WoLF technique.

Results

In this section we will apply WoLF-PHC on different sets and display the results. In order to observe the convergence of WoLF, we have the two matrix sets shown in figure 1. And we will also apply on two general-sum games, a GridWorld and a Soccer.

We will therefore launch training sessions of players using the same algorithms and compare them. PHC and WoLF being rational algorithms, they are supposed to converge by playing against themselves. For matrix games we will choose $\delta_l and \delta_w$ such that $\delta_l = 2 * deltaw$. For the other games we will take more aggressive values with $\delta_l = 4 * \delta w$.

Matrix games

This part will show us how the algorithms learn and converge to the Nash equilibrium of their game. The results of the Matching Pennies using the PHC policy are shown in the figure 4.

We can see that both agents oscillate around the equilibrium without really reaching it, with an amplitude that does not really seem to decrease. As for WoLF-PHC, we can observe in the figure 5 that the results converge almost directly and oscillate very close to the equilibrium.

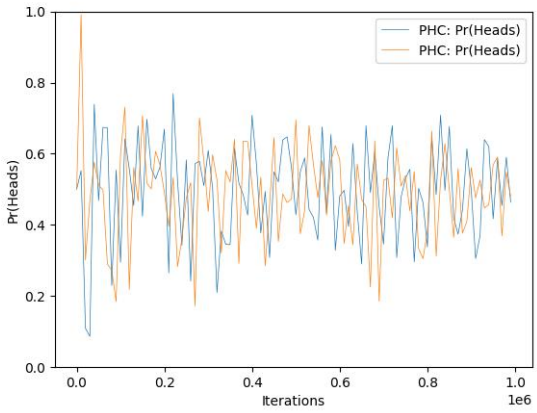

Figure 4: Matching Pennies with PHC policy

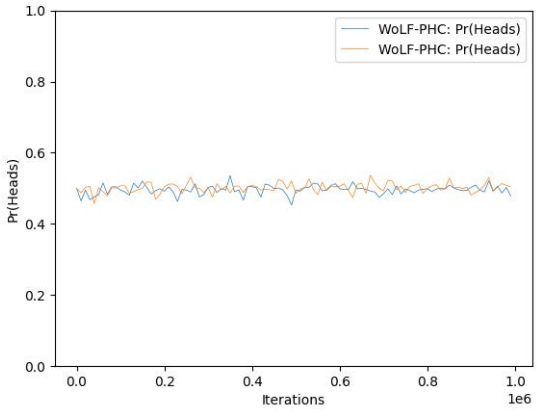

Figure 5: Matching Pennies with WoLF-PHC policy

Let us now observe the results for Rock-Paper-Scissors, in the figure 6, the results seem to rotate without any real rule, which clearly indicates a very slow convergence around the equilibrium.

In contrast to Wolf-PHC, in the figure 7, which despite the semblance of chaos, still remains in a restricted area [0.27, 0.40] and seems to converge much better to equilibrium.

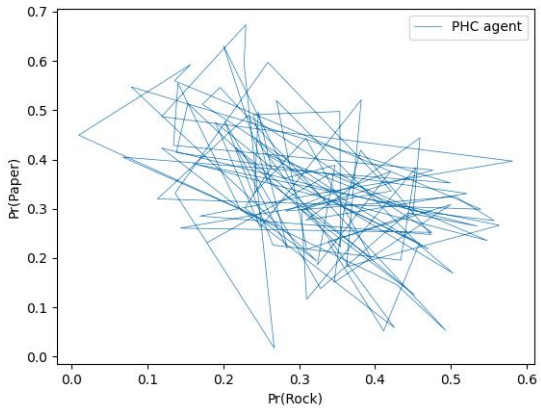

Figure 6: Rock-Paper-Scissors with PHC policy

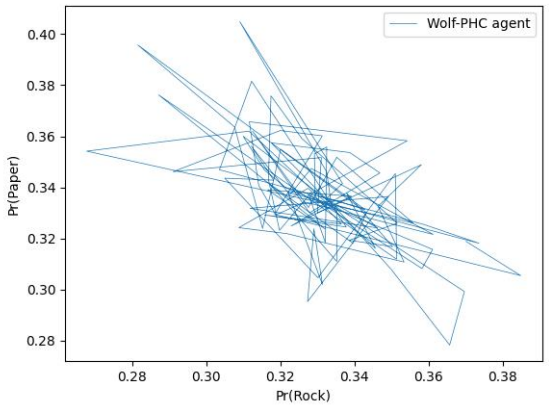

Figure 7: Rock-Paper-Scissors with WoLF-PHC policy

GridWorld

The GridWorld game consists of two players competing on a grid where they must reach a certain cell, both players start in the lower corners of the grid. This grid is represented in the figure 8

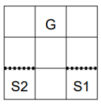

Figure 8: GridWorld Environment

There are two important things to know. The first is that the dotted line indicates a complicated path, in fact, by choosing to go north, the player has only a 50% chance of moving to the cell above, otherwise he does not move.

The second is that if both agents target the same cell, no one moves.

Knowing this, the agents must coordinate the movements and make sure that one chooses the North, and the other the lateral direction.

In the figure 9, we can observe what we have just explained. Player 1 is initially at the bottom left cell of the grid, and will favor his lateral move, the east. His probability of choosing north will therefore decrease accordingly. Another consequence is that player 2, who starts at the bottom right cell, will change his probability to always choose north despite the west direction.

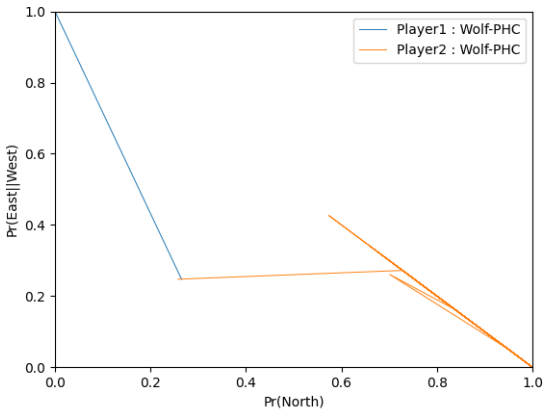

Figure 9: GridWorld Results

This shows that the WoLF-PHC policy converges even in a general-sum game.

Soccer

The last game is the zero-sum game Soccer. The rules are simple and an initial layout is visible in the figure 10.

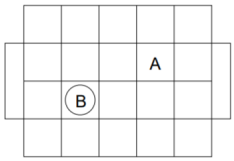

Figure 10: Soccer Environment

Two players compete on a grid, one has a ball, the other does not. The goal of the game is to possess the ball and go into the opponent's goal. If a player moves towards the player who has the ball, he takes the ball from him, without moving. The order of actions is randomly executed each turn, which adds a non-deterministic aspect to the game. The goal here is to train agents to play against themselves for 1 million games. After training, we leave the probabilities alone and have them play against a Q-Learner to see how many games he wins, which will tell us how close the

learning has brought us to the Nash equilibrium of the game which is a mixed policy. The purpose of this is also to compare Littman's Paper[(1)] results with our results.

So we implemented the soccer environment, the Q-Learner and everything we needed to run this simulation. Unfortunately, there was not enough time for us to run the trainings, indeed, we estimated the executions to be several tens of hours counting the hours of debugging/correcting without counting the time put in to perfect the other sections of this article, but the results brought in the initial article seem very reasonable and concrete.

Conclusion

We have seen rationality and convergence, two interesting properties for a multiagent learning algotihm. We have presented an algorithm based on the WoLF principle ("Win or Learn Fast"). This algorithm uses a variable learning rate as indicated in its name. We have detailed all these steps and made connections with the stated properties and proved them with tests on several stochastic games. The algorithm has been shown to be rational while converging to equilibria even when there are several.

References

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163.

Michael., B. and Manuela, V. Rational and convergent learning in stochastic games.