Non-linear Dynamics in Multiagent Reinforcement Learning Algorithms

(Short Paper)

Sherief Abdallah
British University
Dubai, United Arab Emirates
sherief.abdallah@buid.ac.ae

Victor Lesser
University of Massachusetts
Amherst, MA
lesser@cs.umass.edu

ABSTRACT

Several multiagent reinforcement learning (MARL) algorithms have been proposed to optimize agents' decisions. Only a subset of these MARL algorithms both do not require agents to know the underlying environment and can learn a stochastic policy (a policy that chooses actions according to a probability distribution). Weighted Policy Learner (WPL) is a MARL algorithm that belongs to this subset and was shown, experimentally in previous work, to converge and outperform previous MARL algorithms belonging to the same subset.

The main contribution of this paper is analyzing the dynamics of WPL and showing the effect of its non-linear nature, as opposed to previous MARL algorithms that had linear dynamics. First, we represent the WPL algorithm as a set of differential equations. We then solve the equations and show that it is consistent with experimental results reported in previous work. We finally compare the dynamics of WPL with earlier MARL algorithms and discuss the interesting differences and similarities we have discovered.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

Keywords

Reinforcement Learning, Multiagent Systems, Dynamics, Convergence Analysis

1. INTRODUCTION

Our focus in this paper is on a class of MARL algorithms that use a gradient-ascent approach to guide policy search. We will refer to that class as GA-MARL throughout the paper. The general idea of GA-MARL algorithms (more details later in Section 2) is to approximate the policy-gradient using a payoff-gradient and follow the gradient until reaching a local maxima.

A GA-MARL algorithm learns a stochastic policy (a policy that chooses actions according to a probability distribution) without knowing the underlying model of the environment. This ability is particularly important when the world is not fully observable. Another advantage of GA-MARL algorithms is their (relative) simplicity, which makes analyzing their dynamics possible.

The first GA-MARL algorithm whose dynamics were analyzed is the Infinitesimal Gradient Ascent (IGA) algorithm. The dynamics of IGA were linear and IGA's convergence was fairly limited [1]. The IGA-WoLF algorithm had later been developed to address IGA's limitations. The dynamics of IGA-WoLF were piece-wise-linear[1] and IGA-WoLF made strong assumptions in order converge.

We previously developed the Weighted Policy Learner (WPL) [2], which we showed experimentally to converge without knowing the equilibrium strategy, a major improvement over IGA-WoLF. The main contribution of this paper is providing an analysis of WPL's dynamics, showing that it is non-linear, and comparing it to other gradient-based MARL algorithms.

The document is organized as follows. Section 2 introduces previous gradient ascent multiagent learning algorithms. Section 3 breifly described the WPL algorithm. Section 4 formulates the WPL algorithm as a set of differential equations. Section 5 discusses the symbolic solution of WPL's differential equations and how it differs from previous gradient ascent MARL algorithms. In Section 6 we present the results of solving WPL's differential equations numerically, and compare our results to the experimental results reported in previous work. Section 7 discusses WPL's dynamics in comparison to previous gradient-based MARL algorithms. Finally in Section 8 we conclude and discuss future work.

2. GRADIENT-BASED MARL ALGORITHMS

The first gradient-based MARL algorithm whose dynamics were analyzed is the Infinitesimal Gradient Ascent (IGA) [1]. IGA is a simple gradient ascent algorithm where each agent i updates its policy π_i to follow the gradient of expected payoffs (or the value function) V_i. The following equations describe how an agent using IGA updates its policy.

$$\Delta\pi_i \leftarrow \eta\frac{\partial V_i(\pi)}{\partial \pi_i}$$
$$\pi_i \leftarrow limit(\pi_i + \Delta\pi_i)$$

Cite as: Non-linear Dynamics in Multiagent Reinforcement Learning Algorithms (Short Paper), S. Abdallah and V. Lesser, Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1321-1324.
Copyright°c 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

[1] We review the analysis of both IGA and IGA-WoLF dynamics later in the paper.

Variable η is called the learning rate and approaches zero in the limit ($\eta \to 0$) (hence the word Infinitesimal in IGA). Function *limit* projects the updated policy to the space of valid policies, i.e. where $limit(x) = argmin_{x':valid(x')}|x - x'|$.[2] A policy is valid if it sums to 1 and every action is played with non-negative probability.

IGA does not converge in all two-player-two-action games. Algorithm IGA-WoLF (WoLF stands for Win or Learn Fast) was proposed [4] in order to improve convergence properties of IGA by using two different learning rates. More formally,

$$\Delta\pi_i(a) \leftarrow \frac{\partial V_i(\pi)}{\partial \pi_i}(a) * \begin{cases} \eta_{lose} & \text{if } V_i(\pi_i, \pi_{-i}) < V_i(\pi_i^*, \pi_{-i}) \\ \eta_{win} & \text{otherwise} \end{cases}$$
$$\pi_i \leftarrow limit(\pi_i + \Delta\pi_i)$$

Notice that if an agent moves away from its equilibrium policy, this means the value (expected reward) of the current policy is higher than the value of the equilibrium policy and vice versa (which explains the conditions in the above equation). The dynamics of IGA-WoLF have been analyzed and proven to converge in all 2-player-2-action games [4], as we briefly review in the following section. IGA-WoLF has limited practical use, however, because it requires each agent to know its equilibrium policy.

3. WEIGHTED POLICY LEARNER (WPL)

The WPL algorithm is shown in Algorithm 1 for agent i. Variable Δ is the policy gradient that is used to update policy π_i. The idea of the algorithm is to start learning fastest when Δ changes its direction and then to gradually slow down learning if the policy gradient does not change its direction.

Algorithm 1: WPL: Weighted Policy Learner

```
begin
    $\hat{V} \leftarrow$ total average reward $= \frac{\sum_{a \in A_i} V_i(a)}{|A|}$.
    foreach action $a \in A_i$ do
        $\Delta(a) \leftarrow V_i(a) - \hat{V}$
        if $\Delta(a) > 0$ then $\Delta(a) \leftarrow \Delta(a)(1 - \pi_i(a))$
        else $\Delta(a) \leftarrow \Delta(a)(\pi_i(a))$
    end
    $\pi_i \leftarrow limit(\pi_i + \eta\Delta)$
end
```

WPL detects changes in the gradient direction using the difference between action rewards. If the reward of action a is decreasing, then the change in $\pi_i(a)$, $\Delta(a)$, is weighted by $\pi_i(a)$, otherwise it is weighted by $(1 - \pi_i(a))$. Therefore, the largest positive change in $\pi_i(a)$, $\Delta(a)$, is when $\pi_i(a)$ is low and $V_i(a)$ is higher than the average reward $\hat{V}$, and the largest negative change is when $\pi_i(a)$ is near 1 and $V_i(a)$ is lower than $\hat{V}$.

Notice that there are few differences and similarities between IGA-WoLF and WPL's update rules. Both algorithms have two modes of learning rates. IGA-WoLF needs to know the equilibrium strategy in order to distinguish between the two modes, unlike WPL. Also while IGA-WoLF has fixed learning rates for the two modes, WPL uses a continuous spectrum of learning rates, depending on the current policy. It is this particular feature that causes WPL's dynamics to be non-linear, as we discuss in the following section.

4. FORMULATING WPL AS DIFFERENTIAL EQUATIONS

The policies of two agents, p and q, following WPL can be expressed as follows

$$q(t) \leftarrow limit(q(t-1) + \Delta q(t-1))$$
$$p(t) \leftarrow limit(p(t-1) + \Delta p(t-1))$$

where

$$\Delta q(t) = \begin{cases} \eta(1 - q(t))(u_3 p(t) + u_4) & \text{if } u_3 p(t) + u_4 > 0 \\ \eta q(t)(u_3 p(t) + u_4) & \text{if } u_3 p(t) + u_4 < 0 \end{cases}$$

$$\Delta p(t) = \begin{cases} \eta(1 - p(t))(u_1 q(t) + u_2) & \text{if } u_1 q(t) + u_2 > 0 \\ \eta p(t)(u_1 q(t) + u_2) & \text{if } u_1 q(t) + u_2 < 0 \end{cases}$$

We continue derviation of $q(t)$, and similar analysis holds for $p(t)$.

$$\frac{q(t) - q(t-1)}{\eta} =$$
$$\begin{cases} (1 - q(t))(u_3 p(t) + u_4) & \text{if } p(t) > p^* = -u_4/u_3 \\ q(t)(u_3 p(t) + u_4) & \text{if } p(t) < p^* = -u_4/u_3 \end{cases}$$

As $\eta \leftarrow 0$, the equations above become differential:

$$q'(t) = \begin{cases} (1 - q(t))(u_3 p(t) + u_4) & \text{if } p(t) > p^* = -u_4/u_3 \\ q(t)(u_3 p(t) + u_4) & \text{if } p(t) < p^* = -u_4/u_3 \end{cases}$$

Since WPL is a gradient ascent approach, WPL will converge to a deterministic NE if one exists, similar to IGA and IGA-WoLF. This is clear from the gradient definition: a deterministic NE means one action is *always* better than the other, and therefore the gradient direction always points to it leading to eventual convergence [1]. The challenging case is when there is no deterministic NE (the NE is inside the joint policy space). We will therefore focus on this case.

It should be noted that while IGA and IGA-WoLF needed to take the limit function into account, we can safely ignore the limit function while analyzing the dynamics of WPL for 2-player-2-action games. This is due to the way WPL scales the learning rate using the current policy. By the definition of $\Delta p(t)$, a positive $\Delta p(t)$ approaches zero as $p(t)$ approaches one and a negative $\Delta p(t)$ approaches zero as $p(t)$ approaches zero. In other words, as p (or q) approaches 0 or 1, the learning rate approaches zero, and therefore p (or q) will never go beyond the valid period $[0, 1]$. The following section discusses the symbolic solution of these equations.

5. SYMBOLIC SOLUTION

Our goal is to prove that $p(t)$ and $q(t)$ will eventually converge (i.e. in the limit, when $t \to \infty$) to p^* and q^* respectively. To do so, it is enough to show that if one player starts at a policy q^*, then the next time the player returns to q^* the other player will be closer to its NE, and consequently the joint policy will also be a bit closer.

Figure 1 illustrates this point and depicts $p(t)$ and $q(t)$ over a period of time 0 → $T4$, (the figure to the left shows

[2] This general definition of the *limit* function was later developed [3].

policies evolution over time, while the figure to the right shows the joint policy space). If we can prove that over the period $0 \rightarrow T4$ an agent's policy $p(t)$ gets closer to the NE p^*, i.e. $p_{min2} - p_{min1} > 0$ in Figure 1, then by induction the next time period p will get closer to the equilibrium and so on.

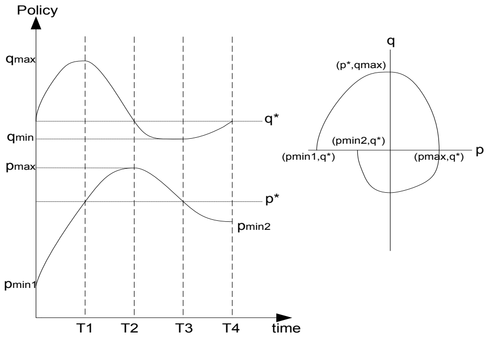

Figure 1: An illustration of WPL convergence.

For readability, p and q will be used instead of $p(t)$ and $q(t)$ for the remainder of this section. The overall period $0 \rightarrow T4$ is divided into four intervals defined by times $0, T1, T2, T3$, and $T4$. Each period corresponds to one combination of $p(t)$ and $q(t)$ as follows. For the period $0 \rightarrow T1$, where $p(t) < p^*$, $q(t) > q^*$: by dividing p' and q'

$$\frac{dp}{dq} = \frac{(1-p)(u_1 q + u_2)}{(1-q)(u_3 p + u_4)}$$

Then by separation we have

$$\int_{p_{min1}}^{p^*} \frac{u_3 p + u_4}{1-p} dp = \int_{q^*}^{q_{max}} \frac{u_1 q + u_2}{1-q} dq$$

$$-u_3(p^* - p_{min1}) + (u_3 + u_4) ln \frac{1 - p_{min1}}{1 - p^*} =$$

$$-u_1(q_{max} - q^*) + (u_1 + u_2) ln \frac{1 - q^*}{1 - q_{max}}$$

Unlike IGA and IGA-WoLF, however, the equations are non-linear and do not have a closed-form solution (note the existence of both x and $ln(x)$). This is the case for the remaining three time periods as well. We solve the equations numerically as described in the following section.

6. NUMERICAL SOLUTION

We used Mathematica and Matlab to solve the equations numerically. Figure 2 shows the theoretical behavior predicted by our model for the matching-pennies game. There is a clear resemblance to the actual (experimental) behavior that was reported in the original WPL paper [2] for the same game (Figure 3). Note that the time-scale on the horizontal axes of both figures are effectively the same, because what is displayed on the horizontal axis in Figure 3 is decision steps. When multiplied by the actual learning rate η used in the experiments, 0.001, both axes become identical.

Figure 4 plots $p(t)$ versus $q(t)$, for a game with NE= $(0.9, 0.9)$ $(u1 = 0.5, u2 = -0.45, u3 = -0.5, , u4 = 0.45)$ and starting from 160 initial joint policies. Figure 6 plots $p(t)$ and $q(t)$ against time, verifying convergence from each of the 160 initial joint policies.

We repeated the above numerical solution for 100 different NE(s) that make a 10x10 grid in the p-q space (starting from the 160 boundary joint policies). The WPL algorithm converges to the NE in a spiral fashion similar to Figure 4 in all the 100 cases. Instead of drawing 100 figures (one for each NE), Figure 5 plots the merge of the 100 figures in a compact way: plotting the joint policy from time 700 to 800 (which is enough for convergence as Figure 6 shows). The two agents converge in all the 100 NE cases, as indicated by the centric points (a diverging algorithm would not have a clean grid with concentrated centric plots).

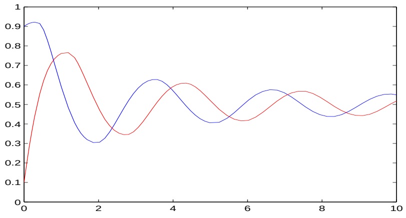

Figure 2: Convergence of WPL as predicted by the theoretical model for the matching pennies game.

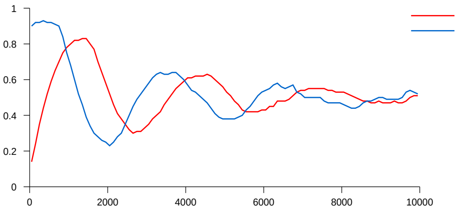

Figure 3: Convergence of WPL through experiments [2].

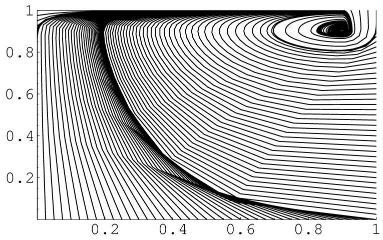

Figure 4: An illustration of WPL convergence to the (0.9,0.9) NE in the p-q space: p on the horizontal axis and q on the vertical axis.

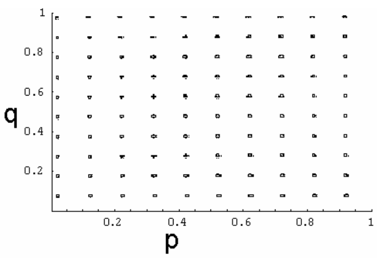

Figure 5: An illustration of WPL convergence for 10x10 NE(s).

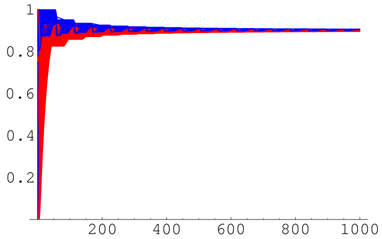

Figure 6: An illustration of WPL convergence to the (0.9,0.9) NE: p(t) (gray) and q(t) (black) are plotted on the vertical axis against time (horizontal axis).

7. COMPARING DYNAMICS OF IGA, IGA-WOLF, AND WPL

With differential equations modeling each of the three algorithms, we now compare their dynamics and point out the main distinguishing characteristics of WPL. Matlab was again used to solve the differential equations (of the three algorithms) numerically. Figure 7 shows the dynamics of the three algorithms in a game with $u1u3 < 0$ and the NE=(0.5,0.5). The joint strategy moves in clockwise direction. The dynamics of WPL are very close to IGA-WoLF, with slight advantage in favor of IGA-WoLF (after one complete round around the NE, IGA-WoLF is closer to the NE than WPL). It is still impressive that WPL has comparable performance to IGA-WoLF, while WPL does not require agents to know their NE strategy a priori.

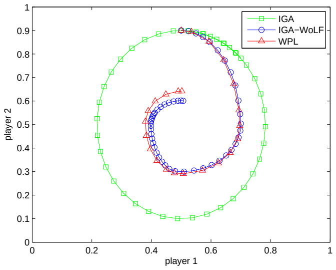

Figure 7: Dynamics of IGA, IGA-WoLF, and WPL in a game with NE=(0.5,0.5).

Figure 8 shows the dynamics in a game with again $u1u3 < 0$ but the NE=(0.5,0.1). Three interesting regions in the figure are designated with A,B, and C. Region A shows that both IGA and IGA-WoLF dynamics are *discontinuous* due to the hard constraints on the policy. Because WPL uses a smooth policy weighting scheme, the dynamics remain continuous. This is also true in region B. In region C, WPL initially deviates from the NE more than IGA, but eventually converges as well. The reason is that because the NE, in this case, is closer to the boundary, policy weighting makes the vertical player move at a much slower pace when moving downward (the right half) than the horizontal player.

Figure 9 shows the dynamics for the coordination game, starting from initial joint policy (0.1,0.6). The coordination game has two NEs: (0,0) and (1,1). All algorithms converge to the closer NE, (0,0), but again we see that both IGA and IGA-WoLF have discontinuity in their dynamics, unlike WPL which smoothly converge to the NE.

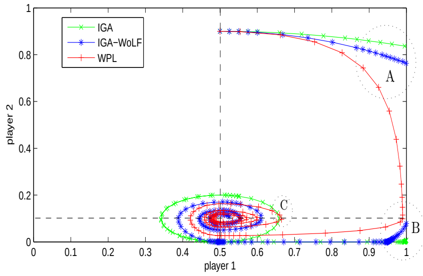

Figure 8: Dynamics of IGA, IGA-WoLF, and WPL in a game with NE=(0.5,0.1).

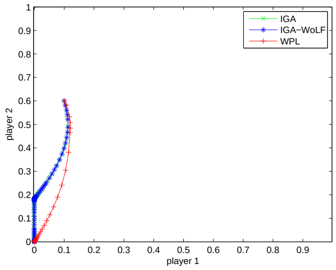

Figure 9: Dynamics of IGA, IGA-WoLF, and WPL in the coordination game with two NEs=(0,0) and (1,1).

8. CONCLUSION AND FUTURE WORK

The main contribution of this paper is formally analyzing the Weighted Policy Learner algorithm and showing that it is the first gradient-ascent (GA) MARL algorithm with non-linear dynamics. The paper models the WPL algorithm for two-player-two-action games as a set of differential equations and then discusses both symbolic and numerical solutions to the equations. The predicted theoretical behavior closely resembles and confirms previously obtained experimental results. Furthermore, the paper solves the equations for 100 games, each starting from 160 initial joint policies and verified WPL's convergence in all of them. Finally, a comparison of WPL's dynamics with previous GA-MARL algorithms' dynamics is given, along with a discussion of similarities and differences.

9. REFERENCES

[1] Singh, S., Kearns, M., Mansour, Y.: Nash convergence of gradient dynamics in general-sum games. In: the 16th Conference on Uncertainty in Artificial Intelligence. (2000) 541–548

[2] Abdallah, S., Lesser, V.: Learning the task allocation game. In: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2006)

[3] Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of the International Conference on Machine Learning. (2003) 928–936

[4] Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence **136**(2) (2002) 215–250