A Common Gradient in Multi-agent Reinforcement Learning

(Extended Abstract)

Michael Kaisers, Daan Bloembergen, Karl Tuyls
Maastricht University, P.O. Box 616, 6200MD, Maastricht, The Netherlands
{michael.kaisers, daan.bloembergen, k.tuyls}@maastrichtuniversity.nl

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

General Terms

Algorithms, Theory

Keywords

Multi-agent learning, Evolutionary game theory, Dynamical Systems, Gradient Learning

1. INTRODUCTION

This article shows that seemingly diverse implementations of multi-agent reinforcement learning share the same basic building block in their learning dynamics: a mathematical term that is closely related to the gradient of the expected reward. Specifically, two independent branches of multi-agent learning research can be distinguished based on their respective assumptions and premises. The first branch assumes that the value function of the game is known to all players, which is then used to update the learning policy based on *Gradient Ascent*. Notable algorithms in this branch include Infinitesimal Gradient Ascent (IGA) [7], the variation Win or Learn Fast IGA (WoLF) [3] and the Weighted Policy Learner [1]. The second branch of multi-agent learning is concerned with learning in unknown environments, using interaction-based *Reinforcement Learning*, and contains those algorithms which have been shown to be formally connected to the replicator equations of *Evolutionary Game Theory*. In this case, the learning agent updates its policy based on a sequence of $\langle action, reward \rangle$ pairs that indicate the quality of the actions taken. Notable algorithms include Cross Learning (CL) [4], Regret Minimization (RM) [6], and Frequency Adjusted Q-learning (FAQ) [5]. This article demonstrates inherent similarities between these diverse families of algorithms by comparing their underlying learning dynamics, derived as the continuous time limit of their policy updates. These dynamics have already been investigated for algorithms from each family separately [1, 2, 3, 5, 6, 7], however, they have not yet been discussed in context of the relation to each other, and the origin of their similarity has not been discussed satisfactorily. In addition to the formal derivation, directional field plots of the learning dynamics in representative classes of two-player two-action games illustrate the similarities and strengthen the theoretical findings.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. ANALYSIS

This section presents an overview of the dynamics of the different algorithms, and highlights their similarities. The discussion is limited to the domain of two-player normal form games for sake of clarity. In these games the payoff function can be represented by a bi-matrix. A general payoff matrix for two-action games, and two specific examples can be denoted as follows, where player 1 and 2 select row i or column j and receive a reward of A_{ij} or B_{ij} respectively:

$$\begin{pmatrix} A_{11}, B_{11} & A_{12}, B_{12} \\ A_{21}, B_{21} & A_{22}, B_{22} \end{pmatrix} \quad \begin{matrix} & C & D \\ C & \frac{3}{5}, \frac{3}{5} & 0, 1 \\ D & 1, 0 & \frac{1}{5}, \frac{1}{5} \end{matrix} \quad \begin{matrix} & H & T \\ H & 1, 0 & 0, 1 \\ T & 0, 1 & 1, 0 \end{matrix}$$

Payoff bi-matrix Prisoner's Dilemma Matching Pennies

The dynamics within two-player two-action games have been studied individually for several algorithms. Let x and y denote the probability of selecting the first action by the first and second player respectively. Furthermore, α is the learning rate, $h = (1, -1)$, V is the value function and x^e is a probability belonging to a Nash equilibrium. FAQ also has a temperature parameter τ that controls the balance between exploration and exploitation. Unifying the notation from literature and factoring out the common gradient $\eth = \left[yhAh^T + A_{12} - A_{22}\right]$ the learning dynamics $\dot{x}$ for player 1 can be summarized as follows:

Alg.	$\dot{x}$ (change in probability of first action)
IGA	$\alpha\eth$
WoLF	$\eth \cdot \begin{cases} \alpha_{min} & \text{if } V(x,y) > V(x^e,y) \\ \alpha_{max} & \text{otherwise} \end{cases}$
WPL	$\alpha\eth \cdot \begin{cases} x & \text{if } \eth < 0 \\ (1-x) & \text{otherwise} \end{cases}$
CL	$\alpha x(1-x)\ \eth$
FAQ	$\alpha x(1-x)\left[\eth \cdot \tau^{-1} - \log \frac{x}{1-x}\right]$
RM	$\alpha x(1-x)\ \eth \cdot \begin{cases} (1+\alpha x \eth)^{-1} & \text{if } \eth < 0 \\ (1-\alpha(1-x)\eth)^{-1} & \text{otherwise} \end{cases}$

More generally, the first and second players' policy will be denoted as π and σ (i.e., $\pi_1 = x$ and $\sigma_1 = y$ in two-action games). The dynamics are illustrated in Figure 1 and are

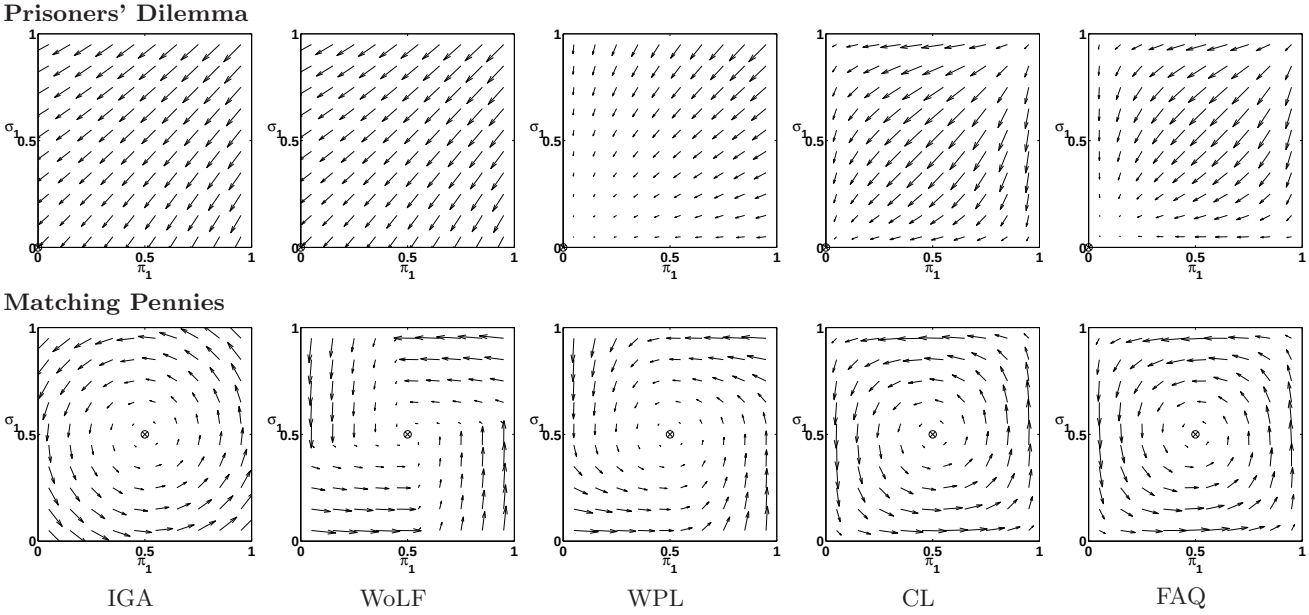

Figure 1: This figure shows the learning dynamics of the various algorithms in the Prisoners' Dilemma and Matching Pennies. The dynamics of RM are visually indistinguishable from CL in this scenario. The Nash Equilibria are indicated with $\otimes$.

now decomposed qualitatively for the class of two-agent normal form games. Let e_i denote the i^{th} unit vector and let n be the number of actions. Gradient Ascent is defined using the orthogonal projection function Φ, which projects the gradient onto the policy simplex thereby ensuring a valid policy (i.e., $\forall \pi_i : 0 \leq \pi_i \leq 1$).

$$\Delta\pi_i \leftarrow \alpha \frac{\partial V(\pi,\sigma)}{\partial \pi_i} = \alpha \lim_{\delta \to 0} \frac{\left[\pi + \Phi(\delta e_i)\right] A\sigma^T - \pi A\sigma^T}{\delta}$$
$$= \alpha\Phi(e_i)A\sigma^T = \alpha\left(e_i A\sigma^T - \frac{1}{n}\sum_j e_j A\sigma^T\right)$$

In contrast, CL, FAQ and RM ensure validity of the policy update by making the update rule proportional to π. Incorporating proportional updating into the gradient-based policy update rule yields $\pi_i(t+1) \leftarrow \pi_i(t) + \pi_i \frac{\partial V(\pi,\sigma)}{\partial \pi_i}$.

In order to incorporate this different approach, the projection function Φ needs to change as well in order to properly map the weighted gradient. Intuitively, this can be achieved by using a weighted mean instead of a standard mean, such that $\hat{\Phi}(\zeta, w) = \zeta - \sum_j w_j \zeta_j$ where w is a normalized weight vector. Using $w = \pi$, this leads to the following dynamics:

$$\dot{\pi}_i = \pi_i \lim_{\delta \to 0} \frac{\left[\pi + \hat{\Phi}(\delta e_i, \pi)\right] A\sigma^T - \pi A\sigma^T}{\delta}$$
$$= \pi_i \hat{\Phi}(e_i,\pi)A\sigma^T = \pi_i[e_i A\sigma^T - \pi A\sigma^T]$$

These resulting dynamics are exactly the dynamics of Cross Learning, showing that it is equivalent to Gradient Ascent with proportional updates. This provides a strong link between the two families of algorithms, gradient ascent on the one hand and independent multi-agent reinforcement learning on the other.

3. CONCLUSIONS

The main contributions can be summarized as follows: First, it is shown that the replicator dynamics are a prime building block of various types of independent reinforcement learning algorithms, such as Cross Learning, Regret Minimization, and Q-learning. Second, the replicator dynamics are shown to relate to the gradient of the expected reward, which forms the basis of Gradient Ascent. Both the replicator dynamics and gradient ascent have an update rule that is based on the difference between the expected reward of an action and the average expected reward over all actions. The only difference is how each action's update is weighted: gradient ascent assumes uniform weights as given by the gradient, whereas the replicator dynamics use the action-selection probabilities as weights.

4. REFERENCES

[1] S. Abdallah and V. Lesser. A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, 33(1):521–549, 2008.

[2] T. Börgers and R. Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1), November 1997.

[3] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[4] J.G. Cross. A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2):239, 1973.

[5] M. Kaisers and K. Tuyls. Frequency adjusted multi-agent Q-learning. In *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315, May, 10-14, 2010.

[6] T. Klos, G. van Ahee, and K. Tuyls. Evolutionary dynamics of regret minimization. *Machine Learning and Knowledge Discovery in Databases*, pages 82–96, 2010.

[7] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 541–548, 2000.