Countering Deception in Multiagent Reinforcement Learning

Bikramjit Banerjee and Jing Peng
Department of Electrical Engineering & Computer Science
Tulane University
New Orleans, LA 70118
{banerjee,jp}@eecs.tulane.edu

ABSTRACT

Multiagent Reinforcement Learning (MRL) is a growing area of research. What makes it particularly challenging is the non-stationarity of the target function. Most of the existing work in this area, however, address either stationary environments or self-play. We assume an asymmetric and non-stationary environment where other agents can be of arbitrary dispositions. In particular, agents can be malicious that model the algorithm being used by the learner and exploit its loopholes. We propose a simple learner that can naively counter such deceptions but at a cost of noise-sensitivity. Then we refine this strategy to maximize its counter deception properties while minimizing its sensitivity to noise.

1. INTRODUCTION

Multiagent Learning (MAL) is an intersection of Distributed Artificial Intelligence (DAI) and Machine Learning (ML). Broad and well-developed as Machine Learning is, it is still incomplete without adequately addressing learning in Multiagent Systems (MAS). Agents in a MAS typically operate in large, complex, open, dynamic and unpredictable environments which forms the key difference between MAL and ML. It is necessary to endow such agents with capabilities to adapt to such environments that basically complement ML. MAL is also an area where ML and Game Theory [13] meet. The latter has been extensively used for modeling concurrent reinforcement learning problems. Several algorithms for multiagent reinforcement learning have been proposed [1, 2, 8, 9, 15], mostly guaranteed to converge to an equilibrium [13] in the limit. Most of these algorithms either assume that the environment is stationary, i.e. the other agents are following stationary policies or they assume some explicit knowledge of the domain [10, 11], or they are explicitly formulated for self-play situations. Littman has explored leader strategies [12] that assume that the opponents can play non-stationary strategies and will, as a result, adapt to the learner's limited policies. There is indeed very little work renouncing any simplifying assumptions about the domain or the opponents or the environment in general.

On the other hand without such assumptions in an open, dynamic and uncertain environment, it is extremely difficult to address rational learning. An agent can be faced with other agents of arbitrary types and it has to be able to predict the other's behavior in order to itself behave optimally. Noting the limiting assumptions in MAL, approaches have even been proposed to exploit them [7]. In this paper we are concerned with a learner potentially facing such exploitative and non-stationary opposition, but itself being unable/reluctant to model them explicitly. As a result the agent is unable to distinguish friends from foe. We ask the question, how should such an agent behave in such environments, when convergence is not a concern? We show that even a naive agent can prevent exploitation by more sophisticated modeling exploiters but it is extremely sensitive to noise in that it is overreactive and could alienate potential friends. We then define a *reward window* for its estimation of action values and show that by adjusting the size of this window, the learner can control the extent of its reactivity. Finally we optimize this window size (leading to an optimal learning rate) to prevent overreaction but at the same time optimizing its capability to counter deception.

2. DEFINITIONS

Here we define some terms and concepts that we will need in the rest of the paper.

Definition 1. *A Stochastic Game with n agents, is a quadruple $\{S, A, T, R\}$, where S is the set of joint states of the agents, A is the set of joint actions avaiable to the agents, T is the joint transition function, $T : S \times A \rightarrow PD(S)$, PD being a probability distribution, and R is the reward function, $R : S \times A \rightarrow \Re^n$.*

A multiagent reinforcement learning task is usually modeled as a stochastic game with $A = A_1 \times A_2 \times \ldots A_n$ where A_i is the set of actions available to the ith agent. The functions T and R are usually unknown. The goal of the ith agent is to find a strategy π^i (a probability distribution over its actions) that maximizes its expected sum of discounted rewards,

$$v(s, \pi_i) = \sum_{t=0}^{\infty} \gamma^t E(r_t^i | \pi_i, \pi_{-i}, s_0 = s)$$

where s_0 is the initial joint-state, r_t^i is the reward of the *ith* agent at time t, $\gamma \in [0, 1)$ is the discount factor, and π^{-i} is the joint strategy of i's opponents.

DEFINITION 2. *A bimatrix game is given by a pair of matrices, (M_1, M_2), (each of size $|A_1| \times |A_2|$ for a two-agent game) where the payoff of the kth agent for the joint action (a_1, a_2) is given by the entry $M_k(a_1, a_2)$, $\forall (a_1, a_2) \in A_1 \times A_2$, $k = 1, 2$.*

Each stage of a stochastic game for two agents (it can be extended to n agents using n-dimensional tables instead of matrices) can be looked upon as a bimatrix game. A *zero-sum game* is a special bimatrix game where $M_1(a_1, a_2) + M_2(a_1, a_2) = 0$, $\forall (a_1, a_2) \in A_1 \times A_2$.

3. COUNTERING DECEPTIVE AGENTS

In a heterogeneous multiagent system there can be agents with varying goals, intentions and dispositions. A self-interested learner can be faced with all kinds of agents and has to learn to optimize its performance against or in collaboration with such agents. The commonest approaches to such problems involve modeling the other agents in some way and trying to predict their behaviors [3, 5, 4, 6, 7, 9]. However, model-building can be a complex task inevitably involving some costly perceptions and/or communications. Oftentimes they are also a potential distraction from main problem solving activities. On the flipside, without some mechanism to predict the other agents' behaviors it may be extremely difficult to achieve the main goal of the learner, viz. maximizing payoff from the environment. In this paper, we focus on the latter task disregarding the actual nature of the opponent. We assume that the learner is either unable (due to design limitations) or reluctant (in order to focus on problem solving) to model the opponents/environment but still wants to be deception-proof as best as it can.

We consider a multiagent reinforcement learning setup involving a single stage, i.e. one shot games, between two agents. We assume that the learner is a naive Q-learner (we call it $\mathcal{Q}$) that maintains a Q-table for the values of its possible actions updating them as

$$Q_{t+1}(a) = (1 - \alpha) Q_t(a) + \alpha r_{t+1} \quad (1)$$

where $\alpha \in (0, 1]$ is the learning rate and r_t is the reward at time step t[1]. The agent's action choice (a_t) at time t is simply the maximizing action, i.e.

$$a_t = \arg\max_b Q_t(b)$$

If more than one action has the same maximum value, then it chooses one of these actions at random. Note that $\mathcal{Q}$ is unable to play mixed strategies, i.e. its strategies π_t^i at all times t are pure - executing only one of the actions with certainty at any time. Since equilibrium of many interesting games consist of mixed strategies, this agent is unable to converge to such equilibria. However, our intention here is not to be able to converge to a game's equilibrium; rather we want to be able to respond optimally to an opponent's variable policies, particularly when such variations are exploiting the learner. As such the explicit goal of the learner is quick adaptability to the environment under the assumption that it is non-stationary. Hence instead of the usual decaying scheme employed for α in reinforcement learning problems [14], the α employed here is a constant which is necessary for non-stationary environments.

[1]The subscript in r_{t+1} simply reflects the fact that the reward is received before the Q-values are updated.

1. Input $\alpha \in (0, 1], \delta \in (0, 1]$. Initialize $Q(s, a) \leftarrow 0$, $\pi(s, a) \leftarrow \frac{1}{|A_i|}$
2. Repeat
   (a) From state s, select action a according to $\pi(s, .)$ with suitable exploration.
   (b) Observe reward r and next state s' and update
   $$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha[r + \gamma \max_b Q(s', b)].$$
   (c) Update
   $$\pi(s, a) \leftarrow \pi(s, a) + \begin{cases} \delta & \text{if } a = \arg\max_b Q(s, b) \\ \frac{-\delta}{|A_i| - 1} & \text{otherwise} \end{cases}$$
   while keeping $\pi(s, .)$ a legal probability distribution.

Table 1: The PHC algorithm

3.1 Sample Performance of $\mathcal{Q}$

The algorithm PHC-Exploiter [7] was developed to exploit a simple Policy Hill Climber (PHC), shown in Table 1. The PHC-Exploiter is a sophisticated modeler designed to exploit the weakness of the opponent's learning algorithm. Its behavior can be characterized by two distinct phases; in one phase it behaves just like the PHC and guides the opponent (PHC) into a region of the latter's policy space where it can exploit (which is the other phase) the opponent's policy and make finite gains. These phase-wise gains over time can add up to unlimited cumulative gains. During the exploitative phase the Exploiter behaves rather like $\mathcal{Q}$. We also note that $\mathcal{Q}$ is a special case of PHC where instead of raising the probability of the maximising action a (step 2c) by a fixed δ, the probability is raised by $1 - \pi(s, a)$. Figure 1 shows the performance of $\mathcal{Q}$ against the PHC-Exploiter for various values of the former's learning rate, α, against a fixed learning rate ($\alpha = 0.9$) of the latter, in Matching Pennies. This is a zero-sum game given by the payoffs

$$M_1 = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}, M_2 = -M_1$$

where M_1 is assumed to be PHC-Exploiter's payoff matrix. This game is particularly suited for studying cumulative payoffs since at each step the payoff increases or decreases by 1. The cumulative reward curves for the Exploiter in Figure 1 indicates that not only does the PHC-Exploiter fail to exploit $\mathcal{Q}$, but also gets punished for opportunistic behavior. It is also evident from Figure 1 that with higher α, $\mathcal{Q}$ can punish Exploiter more sharply. The $\mathcal{Q}$ initializes all the Q-values to 0, thus always playing a uniform random policy for $\alpha = 0$, and this achieves the expected equilibrium payoff of 0, since uniform distribution is indeed the equilibrium policy of this game.

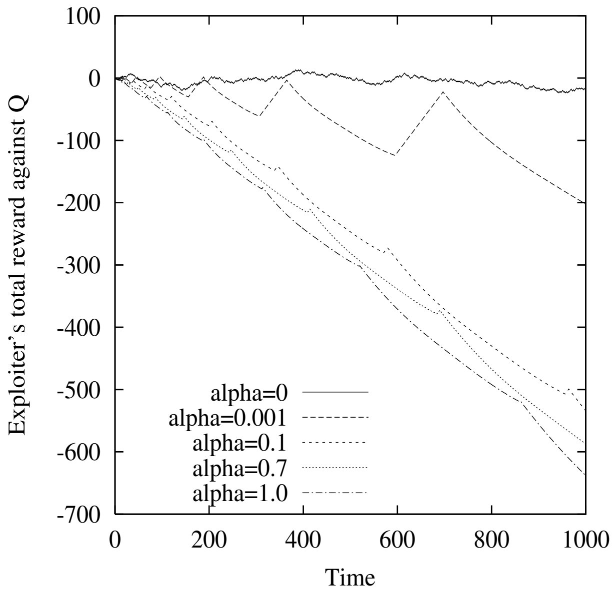

Figure 1: PHC-Exploiter's cumulative rewards against $\mathcal{Q}$ for various values of learning rate parameter of the latter.

3.2 Reactivity: A Reward Window Perspective

We argue that the reason $\mathcal{Q}$ beats Exploiter is the former's better capability to *react* to the opponent's current policy and that α is a measure of this ability of $\mathcal{Q}$ to react to environmental changes. E.g, when $\alpha = 1$ the $\mathcal{Q}$ reacts sharply to the opponent's most recent policy and is unable to learn or generalize. It essentially plays the best response to the opponent's policy consisting of only its last action. When $\alpha \to 0$ the agent plays a fixed policy dictated by its fixed Q-values, and is totally unable to react. For intermediate values, its reactivity falls as α falls from 1 towards 0. This is also corroborated by Figure 1 where higher reactivity of $\mathcal{Q}$ corresponds to lower cumulative payoff curve for the Exploiter and vice-versa.

We note that when $\alpha = 1$, the learner's Q-values reflect just one step of its reward history, while all the values from its history (till the step $t = 0$) are used when $\alpha \to 0$. The update rule in equation 1 actually stores a weighted average of the past rewards received from action a. According to this rule $Q_t(a)$ can be rewritten as

$$Q_t(a) = (1-\alpha)^t Q_0(a) + \sum_{i=1}^{t} \alpha(1-\alpha)^{t-i} r_i \qquad (2)$$

i.e, the most recent reward from action a, r_t, is weighted by α, the previous reward from action a, r_{t-1} is weighted by $\alpha(1-\alpha)$ and in general, the reward from action a, i steps into the past, i.e. r_{t-i}, is weighted by progressively decaying $\alpha(1-\alpha)^i$. This spawns the concept of a reward *window* w, where $w = 1, 2, 3, \ldots$. It is a finite size (w) window stretching from the current time t exactly w steps into the past, within which the rewards are considered significant. This can be constructed by requiring that $(1-\alpha)^w = \delta$, a pre-defined truncation constant, such that $(1-\alpha)^{w+k}, k = 1, 2, \ldots$ are insignificant with respect to δ and the rewards of the corresponding time periods are ignored[2] in the estimate in equation 2. Obviously the first term in equation 2 being even smaller for $t > w$, is also ignored[3]. Hence in terms of w for a small enough δ,

$$Q_t(a) \approx \sum_{i=t-w}^{t} \alpha(1-\alpha)^{t-i} r_i$$

Since the decay factor is $(1-\alpha)$, if we have a smaller w thus requiring that the series decay faster to reach δ within the smaller w, then we must have a higher α. Figure 2 displays two series of weights, $\alpha(1-\alpha)^{10-i}$ used at $t = 10$ for $w = 5$ and $w = 10$, with $\delta = 0.001$. Note the different values of α (value of the plot at $t = 10$) required for the two series. In

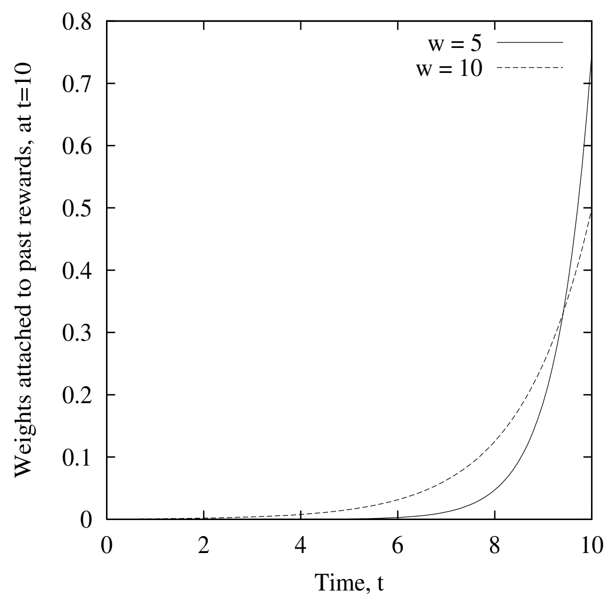

Figure 2: Two series of weights for two values of w for a given $\delta = 10^{-3}$ to be used for $Q_t(a)$ at $t = 10$.

particular if $w = 1$ then we need $\alpha = 1$, i.e. the situation of maximum reactivity. As w increases, more and more of the history is preserved leading to better generalization which is in keeping with the simultaneous lowering of α which in turn corresponds to convergence in classical reinforcement learning. The latter situation also corresponds to lower reactivity, since the Q-estimates change more slowly due to a lower α. Thus w serves as the direct connection between the learning rate parameter α, and the notion of reactivity, for $\mathcal{Q}$.

3.3 Optimum Window

The question arising at this point is what value of w is desirable in a non-stationary environment? With $w = 1$ the agent displays knee-jerk reaction since it immediately throws away almost whole of its history. This can be counterproductive if the sensors are noisy or the learner misreads the

[2] We assume the rewards are bounded so that when the weights are insignificant, the corresponding terms in equation 2 can be ignored.

[3] It can also be ignored if the intial Q-values were 0, i.e. $Q_0(a) = 0$

rewards, since once thrown away the past rewards cannot be brought back. Also the opponent might be exhibiting noisy behavior and not actually intended to exploit it, but might be alienated due to the knee-jerk behavior. Hence too low values of w cannot be desirable. In the same vein, if w is too high then the learner is too slow to adapt to its opponents who may be exploiting the learner with impunity. Consequently, we need to look for an intermediate value of w that will maximize reactivity (which means lower w) while maximizing w itself. We can achieve exactly that by maximizing the quantity $D = \alpha - \frac{1}{w}$ when D is positive. Below we state and prove the necessary condition for D to be positive.

Proposition 1. *It is necessary (but not sufficient) that* $\delta \leq \frac{1}{e}$ *for* $D = \alpha - \frac{1}{w}$ *to be positive.*

Proof : Since $(1-\alpha)^w = \delta$ or $\alpha = 1 - \delta^{\frac{1}{w}}$, we have

$$D = \alpha - \frac{1}{w} = 1 - \delta^{\frac{1}{w}} - \frac{1}{w}$$

For this to be positive, we need

$$\begin{aligned} \delta^{\frac{1}{w}} &\leq 1 - \tfrac{1}{w} \\ \Rightarrow \quad \delta &\leq (1 - \tfrac{1}{w})^w \end{aligned}$$

The maximum value of the right hand side for any admissible w, i.e. $w \in I^+ - \{0\}$, is $\frac{1}{e}$. Hence proved.

For small enough δ, it can be guaranteed that $D = 1 - \delta^{\frac{1}{w}} - \frac{1}{w}$ is positive for all values of $w = 1, 2, \ldots$. Then by maximizing D we get the highest possible reactivity with the lowest possible value of $\frac{1}{w}$, i.e. highest possible value of w. The following proposition establishes the value of w that does this.

Proposition 2. *When* δ *is small enough, the difference* $D = \alpha - \frac{1}{w}$ *is maximized for a window size* $w^* = \frac{\log(\frac{1}{\delta})}{\log\log(\frac{1}{\delta})}$.

Proof : The curves of $D = \alpha - \frac{1}{w}$ for some values of δ are shown in Figure 3. Differentiating $\alpha - \frac{1}{w} = 1 - \delta^{\frac{1}{w}} - \frac{1}{w}$ w.r.t w and setting it to 0 (for w^*), we have

$$\begin{aligned} \delta^{\frac{1}{w^*}} \log(\delta) + 1 &= 0 \\ \Rightarrow \quad \delta^{\frac{1}{w^*}} &= \frac{1}{\log(\frac{1}{\delta})} \\ \Rightarrow \quad \frac{1}{w^*} &= \frac{\log(\frac{1}{\log(\frac{1}{\delta})})}{\log(\delta)} \\ \Rightarrow \quad w^* &= \frac{\log(\frac{1}{\delta})}{\log\log(\frac{1}{\delta})} \end{aligned}$$

The value of w^* corresponds to a learning rate of $\alpha^* = 1 - \frac{1}{\log(\frac{1}{\delta})}$ which gives the optimal learning rate that enables the learner to react to a dynamic environment as best as one can while being careful not to overtrust the most recent payoffs in shifting strategies.

4. CONCLUSIONS & FUTURE WORK

In this paper we have addressed the issue of how a reinforcement learner can tackle a non-stationary environment. We have highlighted the need of a capability to react to environmental changes in a non-stationary multiagent setting and how it can be achieved. This is crucial in ensuring that the learner does not get exploited by other agents in the domain. Our main emphasis in this regard is on a need to balance this capability with noise-sensitivity which could be raised due to a high reactivity. We have theoretically derived an optimal learning rate that maximizes reactivity while minimizing noise-sensitivity. This is the best a learner can do if it is unaware of the nature of the other agents in the domain, or if it is incapable of modeling them. However, if the agent does indeed model the others, then it may be possible to use this strategy in dynamically adapting the learning rate (possibly independently for each action) based on variations of δ according to its estimate of the extent to which the opponents may be exploiting it. This would be an interesting avenue for further exploration. In future, we also intend to compare this approach with model-based approaches to countering deception and establish its cost-effectiveness.

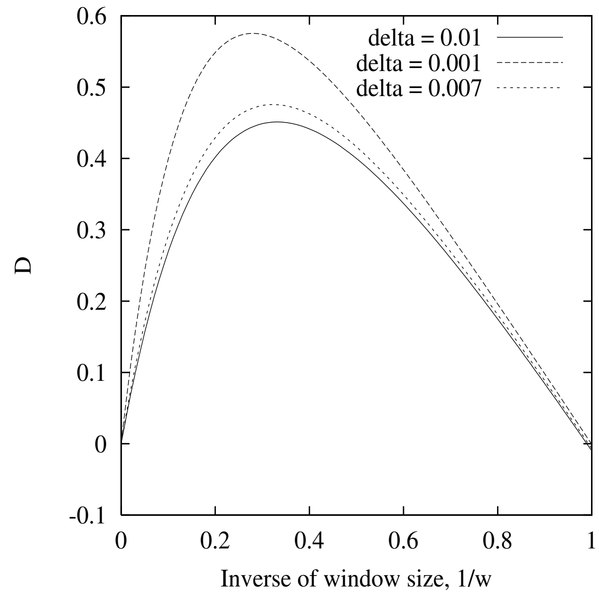

Figure 3: The difference $D = 1 - \delta^{\frac{1}{w}} - \frac{1}{w}$ as a function of $\frac{1}{w}$ for various δ.

5. REFERENCES

[1] B. Banerjee, S. Sen, and J. Peng. Fast concurrent reinforcement learners. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001.

[2] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 2002. In Press.

[3] David Carmel and Shaul Markovitch. Incorporating opponent models into adversary search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 120–125, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[4] David Carmel and Shaul Markovitch. Learning and using opponent models in adversary search. Technical Report Technical Reort CIS report 9609, Technion, 1996.

[5] David Carmel and Shaul Markovitch. Learning models of intelligent agents. In *Thirteenth National Conference on Artificial Intelligence*, pages 62–67, Menlo Park, CA, 1996. AAAI Press/MIT Press.

[6] David Carmel and Shaul Markovitch. Opponent modeling in multi-agent systems. In Gerhard Weiß and Sandip Sen, editors, *Adaptation and Learning in Multi-Agent Systems*, pages 40–52. Springer-Verlag: Heidelberg, Germany, 1996.

[7] Yu-Han Chang and Leslie Pack Kaelbling. Playing is believing: The role of beliefs in multi-agent learning. In *Neural Information Processing Systems*, Vancouver, Canada, 2001.

[8] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 746–752, Menlo Park, CA, 1998. AAAI Press/MIT Press.

[9] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proc. of the 15th Int. Conf. on Machine Learning (ML'98)*, pages 242–250, San Francisco, CA, 1998. Morgan Kaufmann.

[10] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of the 11th Int. Conf. on Machine Learning*, pages 157–163, San Mateo, CA, 1994. Morgan Kaufmann.

[11] M. L. Littman. Friend-or-foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learnig*, Williams College, MA, USA, 2001.

[12] M. L. Littman and P. Stone. Leading best response strategies in repeated games. In *17th International Joint Conference on AI (IJCAI) workshop on Economic Agents, Models and Mechanisms*, 2001.

[13] G. Owen. *Game Theory*. Academic Press, UK, 1995.

[14] R. Sutton and A. G. Burto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[15] Xiaofeng Wang and Tuomas Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Advances in Neural Information Processing Systems 15, NIPS*, 2002.