Efficient No-regret Multiagent Learning

Bikramjit Banerjee and **Jing Peng**
Dept. of Electrical Engineering & Computer Science
Tulane University
New Orleans, LA 70118
{banerjee, jp}@eecs.tulane.edu
http://www.eecs.tulane.edu/{Banerjee, Peng}

Abstract

We present new results on the efficiency of no-regret algorithms in the context of multiagent learning. We use a known approach to augment a large class of no-regret algorithms to allow stochastic sampling of actions and observation of scalar reward of only the action played. We show that the average actual payoffs of the resulting learner gets (1) close to the best response against (eventually) stationary opponents, (2) close to the asymptotic optimal payoff against opponents that play a converging sequence of policies, and (3) close to at least a dynamic variant of minimax payoff against arbitrary opponents, with a high probability in polynomial time. In addition the polynomial bounds are shown to be significantly better than previously known bounds. Furthermore, we do not need to assume that the learner knows the game matrices and can observe the opponents' actions, unlike previous work.

Introduction

Multiagent or concurrent learning is a challenging problem. There are two major sources of uncertainty in concurrent learning domains; the uncertainties in sensing and actuation, and the uncertainties due to the changing behaviors of the other learning agents. Whereas various approaches like reinforcement learning, POMDPs address the first issue, Game Theory addresses the second. This work belongs to a line of research that addresses mainly the second kind of uncertainty and draws inspiration from Game Theory. We also address part of the first kind of uncertainty since we renounce the assumption that a learner needs to observe the opponents' actions.

Recent research in Multiagent Reinforcement Learning (MARL) has seen a markedly greater focus on the performance of concurrent learners than on stability in their behaviors. A latest work (Powers & Shoham 2005) proposed a new set of criteria for MARL agents and devised a Metastrategy to achieve these efficiently. In particular, they guarantee that a learner will achieve average payoff that is

Property 1: near best response against stationary players,

Property 2: close to the payoff of an equilibrium that is not Pareto dominated by another equilibrium, in self-play, and

Property 3: close to the minimax payoff against all other players,

each with probability $1-\delta$ in time polynomial in

$$(1/\delta, K)$$

and other parameters, where K is the number of joint actions of the players. If the number of agents in the domain is n and the size of the action space of each is k, then $K = O(k^n)$. Moreover, Metastrategy needs to assume that each learner

Assumption 1: knows its own game matrix,

Assumption 2: knows the opponents' game matrices,

Assumption 3: can observe the actions of its opponents,

Assumption 4: can observe its own payoffs.

We believe that such strong assumptions are characteristic of meta-level reasoning and may be unnecessary with a different approach. In this paper we take a direct adaptive approach that significantly improves the polynomial bounds and at the same time removes assumptions 1, 2 and 3.

Using an approach from (Auer *et al.* 1998), we provide an augmentation to any of a large class of no-regret algorithms that allows them to play actions instead of mixed policies and observe only rewards of actions played (as is usual in reinforcement learning) rather than the expected payoffs of every action in each round. Our contribution is to show that this version of most no-regret algorithms achieves average payoff that is (1) near best response against (eventually) stationary players and (2) close to at least a dynamic (stronger) variant of minimax payoff against arbitrary opponents, (i.e., Property 1 and an improved version of Property 3 above) in time polynomial in

$$(\ln(1/\delta), k \ln k)$$

besides other parameters, which are significant improvements over Metastrategy's guarantees. In particular our bounds are independent of the number of players (n) in the game. Moreover to achieve these two properties we only need Assumption 4 while Metastrategy uses all 4 assumptions.

We do not explicitly handle Property 2, i.e., self-play. However we do prove, additionally, that if the opponents play a *converging sequence* of joint policies, then our learner will get close to its maximal possible asymptotic payoff quickly. We argue that this indirectly addresses self-play in the following way. If we choose a no-regret algorithm (base algorithm) that is known (or can be shown) to converge in policies in self-play to an equilibrium profile, and if they can choose this profile to be non-Pareto dominated, then our last result will mean that they get close to such an equilibrium payoff quickly. This is equivalent to Property 2 of Metastrategy, but again with improved dependence on δ and k. On the flipside, we will possibly need to bring back assumptions 1 thru 3 depending on which base algorithm we choose.

Multiagent Reinforcement Learning

A Multiagent Reinforcement Learning task is usually modeled (Littman 1994) as a Stochastic Game (SG, also called *Markov Game*), which is a Markov Decision Process with multiple controllers. We focus on stochastic games with a single state, also called repeated games. This refers to a scenario where a matrix game (defined below) is played repeatedly between a learner and its opponents. We call the set of actions available to the learner A, and the set of joint actions of its $n-1$ opponents B. Let $k = |A|$ and potentially all agents have k actions available, so $|B| = O(k^{n-1})$.

Definition 1 *A matrix game for a learner is given by a $|A| \times |B|$ matrix, $\mathbf{R}$, of real values, such that if the learner chooses action $a \in A$ and the opponents choose a joint action profile $b \in B$, then the payoff of the learner will be $R(a, b)$.*

Usually each agent will have a different matrix for its own payoffs, i.e., ith agent will have its own $\mathbf{R}_i$. A *constant-sum game* (a.k.a competitive games; useful in describing two agent competitive interactions) is a special matrix game where for every joint action profile chosen by all agents, the sum of their payoffs is a constant. If this constant is zero, then it is also called a zero-sum game. We assume (as in many previous work) that payoffs are bounded, $R(a, b) \in [0, r_{\max}]$, for real $r_{\max}$. Table 1 shows an example matrix game for two agents, the Shapley game, with $r_{\max} = 1$. Each agent has 3 available actions.

Table 1: The Shapley Game.

$$\mathbf{R}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{R}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

A *mixed policy* is a probability distribution over an agent's action space. We will represent a mixed policy of the learner as vector $\boldsymbol{\pi} \in \Delta(A)$, and a mixed joint policy of its opponents as the vector $\boldsymbol{\rho} \in \Delta(B)$. Here Δ stands for the set of probability distributions. If the entire probability mass of a mixed policy is concentrated on a single action j, then it is also called a *pure policy* represented as the vector $\boldsymbol{\delta}_j$. When the learner plays a mixed policy $\boldsymbol{\pi}$ and its opponents play a mixed policy $\boldsymbol{\rho}$, the learner's expected payoff is given by

$$V(\boldsymbol{\pi}, \boldsymbol{\rho}) = \sum_{a \in A, b \in B} \pi(a)\rho(b)R(a, b).$$

Definition 2 *For an n-player matrix game, the best response of the learner to its opponents' joint policy ($\boldsymbol{\rho}$) is given by $BR_{\boldsymbol{\rho}} = \{\boldsymbol{\pi} | V(\boldsymbol{\pi}, \boldsymbol{\rho}) \geq V(\boldsymbol{\pi}', \boldsymbol{\rho}),\ \forall \boldsymbol{\pi}' \in \Delta(A)\}$.*

Definition 3 *The regret of a learner playing a sequence of policies $\{\boldsymbol{\pi}^t\}_{t=1}^{t=T}$, relative to any policy $\boldsymbol{\pi}$, written as $\mathcal{R}^T(\boldsymbol{\pi})$ is given by*

$$\mathcal{R}^T(\boldsymbol{\pi}) = \sum_{t=1}^{t=T} V(\boldsymbol{\pi}, \boldsymbol{\rho}^t) - \sum_{t=1}^{t=T} V(\boldsymbol{\pi}^t, \boldsymbol{\rho}^t). \tag{1}$$

If the sum (over $t = 1, \ldots, T$) of expected payoffs of the learner against the actual unknown policies played by the opponent were compared to that of an arbitrary policy $\boldsymbol{\pi}$ of the learner, then the difference would be the learner's regret. In hindsight he finds that always playing $\boldsymbol{\pi}$ instead of the sequence $\{\boldsymbol{\pi}^t\}$ would have yielded a total payoff higher than his actual payoff by $\mathcal{R}^T(\boldsymbol{\pi})$ (possibly negative).

Related Work

Multiagent Reinforcement Learning has produced primarily two types of algorithms. One type learns some fixed point of the game e.g., Nash equilibrium (Minimax-Q (Littman 1994), Nash-Q (Hu & Wellman 1998), FFQ (Littman 2001)) or correlated equilibrium (CE-Q (Greenwald & Hall 2002)). These algorithms can guarantee a certain minimal expected payoff asymptotically, but it may be possible to guarantee higher payoff in certain situations if the learner is adaptive to the opponents' play, instead of learning the game solution alone. This brings us to the other type of learners that learn a best response to the opponents' actual play e.g., IGA (Singh, Kearns, & Mansour 2000), WoLF-IGA (Bowling & Veloso 2002), AWESOME (Conitzer & Sandholm 2003). They assume that the opponents are stationary, or equivalently, learn best response to the empirical distribution of the opponents' play. WoLF-IGA and AWESOME also converge to some equilibrium profile in self-play thus guaranteeing convergence of payoffs as well. Simple Q-learning (Sutton & Burto 1998) is also capable of learning a best response to an arbitrary opponent's policy provided that latter is stationary. There has also been some work on playing team games (where the game matrices of all agents are identical) (Claus & Boutilier 1998; Wang & Sandholm 2002) with stronger convergence guarantees owing to the correlation of the game matrices. Most of these convergence results are in the limit.

One significant line of work that is being increasingly explored recently in MAL is on *regret matching* learners. Algorithms have been proposed that achieve $\lim_{T \to \infty} \frac{\mathcal{R}_i^T(\boldsymbol{\pi}_i)}{T} \leq 0$ (called *no-regret* algorithms) for any policy $\boldsymbol{\pi}_i$ (Auer *et al.* 1998; Fudenberg & Levine 1995; Freund & Schapire 1999; Littlestone & Warmuth 1994). Their convergence properties were studied in self-play and found to be incomplete (Jafari *et al.* 2001), but with additional information, no-regret learning was found to be convergent (Banerjee & Peng 2004). These algorithms usually provide guarantees about asymptotic average expected payoffs but little is known about their efficiency or about their average actual payoffs in general. Recent work by Zinkevich shows that a generalized version of IGA called GIGA (Zinkevich 2003) has a no-regret property. Similarly a generalized version of WoLF-IGA called GIGA-

WoLF (Bowling 2005) has no-regret property with additional policy convergence guarantees against GIGA in small games. However, these recent algorithms also lack efficiency guarantees as the other no-regret algorithms.

The one work that our paper is most related to (Powers & Shoham 2005) proposed a new set of properties for a MAL algorithm, with a greater focus on payoff and efficiency than policy convergence. We devise a similar learning strategy based on no-regret algorithms and show that it can provide efficiency guarantees (for at least 2 of their 3 properties), that improve significantly upon (Powers & Shoham 2005) with significantly less assumptions. For the remaining case, we provide evidence of it being satisfied but possibly at the cost of similar (large) set of assumptions.

Augmenting a No-regret Algorithm

Usually a no-regret learner observes at each round, a reward vector that specifies the expected reward of each action if the mixed policy is played (e.g., GIGA (Zinkevich 2003), GIGA-WoLF (Bowling 2005), **Hedge** (Auer *et al.* 1998) etc) and outputs a new policy vector, $\boldsymbol{\pi}^{t+1}$. Let the reward vector that the learner observes for his current policy be $\hat{\mathbf{x}}^t$. Usually $\boldsymbol{\pi}^{t+1}$ is computed from $\hat{\mathbf{x}}^t$ and $\boldsymbol{\pi}^t$ alone, and no extra knowledge (whether about the game or the opponents) is necessary. An augmenting setup was described in (Auer *et al.* 1998) to extend **Hedge** to allow stochastic sampling of actions and observation of the reward for only the action played (**Exp3**). This allowed the authors to prove that the average actual payoffs of the learner can be close to playing the best action with a minimal probability, i.e., lower bounding the per trial gain of the algorithm probabilistically. Actually this strategy can be used in a straightforward way to augment any no-regret algorithm as we show below.

In this paper we consider the class of no-regret algorithms that have $\mathcal{R}^T(\boldsymbol{\pi}) \leq p\sqrt{T} + q$ for any policy $\boldsymbol{\pi}$ and for real values of p, q and $p > 0$. Usually p, q are polynomials of k and $r_{\max}$. E.g., for **Hedge** $p = \sqrt{2\log k}, q = 0$, for GIGA $p = (1 + k.r^2_{\max}), q = -k.r^2_{\max}/2$. Most no-regret algorithms satisfy this form, so the results developed in this paper apply to a large class of no-regret algorithms. Now $V(\boldsymbol{\pi}^t, \boldsymbol{\rho}^t) = \sum_j \pi^t_j \hat{x}^t_j$. As a result of this and the assumed upper bound on regret of this class of no-regret algorithms, we have from equation 1, for any algorithm of this class

$$\sum_{t=1}^{t=T}\sum_{j=1}^{j=k} \pi^t_j \hat{x}^t_j \geq \sum_{t=1}^{t=T}\sum_{j=1}^{j=k} \pi_j \hat{x}^t_j - (p\sqrt{T} + q) \qquad (2)$$

for an arbitrary policy $\boldsymbol{\pi}$. Note that the set of policies that maximize $\sum_t \sum_j \pi_j \hat{x}^t_j$ consists of at least one pure policy, i.e., the action which if always played would have given this maximum payoff sum over t. Therefore the above inequality can be written as

$$\sum_t \sum_j \pi^t_j \hat{x}^t_j \geq \sum_t \hat{x}^t_i - (p\sqrt{T} + q)$$

for all $i = 1 \ldots k$. The stochastic augmented module samples the distribution

$$\hat{\boldsymbol{\pi}}^t = (1-\gamma)\boldsymbol{\pi}^t + \frac{\gamma}{k}\mathbf{1}, \qquad (3)$$

where γ [1] is the probability of exploration, and plays action j_t at time t. It receives the scalar reward x_{j_t} for executing that action and returns to the no-regret algorithm the reward vector $\hat{\mathbf{x}}^t$ given by

$$\hat{x}^t_j = \begin{cases} \frac{x_{j_t}}{\hat{\pi}^t_{j_t}} & \text{if } j = j_t \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

The above compensation scheme for generating a reward vector based on scalar observation ensures that the *expected* gain (i.e. reward aggregate) of any action is proportional to its actual gain. We call this augmented scheme NoRA (No-Regret with Augmentation). It represents a large class of algorithms and all our results apply to all of them. We call any agent following any algorithm from this class, a NoRA agent or a NoRA learner.

Now since $\sum_j \pi^t_j \hat{x}^t_j = \pi_{j_t}\frac{x_{j_t}}{\hat{\pi}_{j_t}} \leq \frac{x_{j_t}}{1-\gamma}$ (from equations 3,4), we have from equation 2

$$\sum_t x_{j_t} \geq (1-\gamma)\sum_t \hat{x}^t_i - (1-\gamma)(p\sqrt{T} + q), \ \forall i.$$

If $\sum_t x^t_i$ is the actual gain of action i thru T, then from lemma 5.1 of (Auer *et al.* 1998) [2] we know that for $1 > \lambda, \delta > 0$ and for any i, with probability at least $1 - \delta/2$

$$\sum_t \hat{x}^t_i \geq \left(1 - \frac{k\lambda}{\gamma}\right)\sum_t x^t_i - \frac{r_{\max}\ln(2k/\delta)}{\lambda}$$

This is a slightly weaker bound than (Auer *et al.* 1998) but is sufficient for our purpose. It is also slightly more general because unlike (Auer *et al.* 1998), we do not need to assume $r_{\max} = 1$. Combining the last two inequalities we have for all $i = 1, \ldots, k$, with probability at least $1 - \delta/2$,

$$\begin{aligned}\sum_t x_{j_t} \quad \geq \quad & \sum_t (1 - \frac{k\lambda}{\gamma} - \gamma)x^t_i - \frac{(1-\gamma)r_{\max}\ln(2k/\delta)}{\lambda} \\ & -(1-\gamma)(p\sqrt{T} + q) \qquad (5)\end{aligned}$$

This is a basic property of any NoRA agent. At this point there is insufficient information to provide tight bounds on the regret. With further algorithm-specific assumptions, **Exp3** achieves $O(T^{\frac{2}{3}})$ regret bound. There is also current interest in gradient estimation (as in equation 4) in partial information settings. For instance, (Flaxman, Kalai, & McMahan 2005) provide an alternate gradient estimation procedure based on scalar feedback similar to our setup, which allows a regret bound of $O(T^{\frac{3}{4}})$ for the corresponding version of GIGA. However, our purpose is not to tightly bound the regret; rather we wish to establish opponent-dependent polynomial time bounds on the accumulated rewards for which equation 5 is a sufficient starting point. Using it we show in the following sections, that the average actual rewards of

[1] The addition of a uniform exploration component is necessary to obtain good actual payoffs in any single run, in contrast with just obtaining good payoffs *in expectation*.

[2] The proof of Lemma 5.1 does not depend on the specifics of **Hedge**, only those of the augmented module of **Exp3**, which we adopt in a general way.

any NoRA learner will satisfy Property 1,3 and also be close to the the maximum possible asymptotic payoff against converging opponents, with a high probability in polynomial time.

(Eventually) Stationary Opponents

Eventually stationary opponents are those that play stationary policies after some finite (but unknown) time t_0. If an algorithm is guaranteed to attain near best response against opponents that are always stationary in polynomial time, then it should not be harder to guarantee the same against eventually stationary opponents with at most an extra polynomial dependence on t_0. Hence in the following, we consider stationary opponents. Let $\boldsymbol{\rho}$ be the fixed joint distribution of the opponents, and V_{BR} be the value of the learner's best response.

Theorem 1 *For $\epsilon > \frac{2\gamma r_{\max}}{\gamma+1/2}, \delta > 0$, there exists a T_0 polynomial in $(1/\epsilon, \ln(1/\delta), r_{\max}, k \ln k)$ such that if the game is played for at least T_0 rounds, a NoRA agent will achieve an average actual payoff of at least $V_{BR} - \epsilon$ against $n-1$ stationary players, with probability at least $1-\delta$.*

Proof : If the opponents are sampling actions from fixed distributions, then the x_i^t's in RHS of equation 5 are i.i.d with mean V_{BR} if $\boldsymbol{\delta}_i \in BR_{\boldsymbol{\rho}}$. By Azuma's Lemma $P(\frac{1}{T_1}\sum_{t=1}^{t=T_1} x_i^t \geq V_{BR} - \epsilon/2) \geq 1 - \exp(\frac{-T_1\epsilon^2}{8r_{\max}^2})$. Setting this to be at least $1-\delta/2$, we have $T_1 \geq \frac{8r_{\max}^2 \ln(2/\delta)}{\epsilon^2}$, a polynomial in $(r_{\max}, \ln(1/\delta), 1/\epsilon)$. Also for all $T_2 > 0$, with probability at least $(1-\delta/2)^2 \geq (1-\delta)$, we have from equation 5

$$\begin{aligned}\sum_{t=1}^{t=T_1+T_2} x_{j_t} \quad \geq \quad & (1 - \frac{k\lambda}{\gamma} - \gamma)(T_1+T_2)(V_{BR} - \epsilon/2) \\ & - \frac{(1-\gamma) r_{\max} \ln(2k/\delta)}{\lambda} \\ & -(1-\gamma)(p\sqrt{T} + q)\end{aligned}$$

Since we are free to choose any admissible value for $1 > \lambda > 0$ we let $\lambda = \gamma^2/k$. Taking average from the above, we see it is at least

$$\begin{aligned}(1-2\gamma)(V_{BR} - \epsilon/2) \quad & - \quad \frac{(1-\gamma) r_{\max} k \ln(2k/\delta)}{\gamma^2 (T_1+T_2)} \\ & - \quad \frac{(1-\gamma)(p\sqrt{T_1+T_2} + q)}{T_1+T_2}\end{aligned}$$

Clearly, it is necessary that $\gamma < 0.5$, i.e., majority of the probability mass should be concentrated on exploitation rather than exploration. Actually γ should be set to a fixed small value. Then for $\epsilon > \frac{2\gamma r_{\max}}{\gamma+1/2}$, setting the above expression for average value at least $V_{BR} - \epsilon$, we get a quadratic of the form of $ax^2 - bx - c = 0$ with $a, b, c > 0$, for the minimal value of $\sqrt{T_1+T_2}$. So a real solution exists and the theorem follows for $T_0 = T_1 + T_2$. ∎

The limitation of this result is that it does not hold for *every* $\epsilon > 0$, only for every $\epsilon > \frac{2\gamma r_{\max}}{\gamma+1/2}$. This is the price of never ceasing to explore. We discuss this limitation further in the conclusion section. Also note that the value of T_0 will be proportional to $1/\gamma$; so if γ is held to a small constant (which can be done independently of the game or the opponents) to allow a small ϵ, the constants of the polynomial expression will be large. However, the advantages of this result over Metastrategy are more compelling. The dependence of NoRA is on $\ln(1/\delta)$ instead of $1/\delta$ and $k \ln k$ instead of k^n. [3] Furthermore such a learner does not need to observe the opponents' actions or know its own game matrix, whereas both of these assumptions are needed in (Powers & Shoham 2005) to produce the guarantees against stationary opponents, and all of Assumptions 1-4 are used. NoRA only needs Assumption 4. All these advantages (as well as the limitation due to exploration) also apply to all results developed in the rest of the paper.

Converging Opponents

If the opponents' joint policy converges in Cauchy's sense, then for any given $\epsilon > 0$ there exists a time t_ϵ such that $\forall t_1, t_2 > t_\epsilon$, $\|\boldsymbol{\rho}^{t_1} - \boldsymbol{\rho}^{t_2}\| < \epsilon$. In other words, there exists a region of diameter ϵ in the joint policy space, N_ϵ, such that $\forall t > t_\epsilon$, $\boldsymbol{\rho}^t \in N_\epsilon$. The policy sequence converges to the center of this "ball", but we do not need to know the location of this ball in the policy space, just the fact that it exists. We call $\boldsymbol{\pi}_{BR(\boldsymbol{\rho})}$ any best response policy of the learner to the opponents joint policy $\boldsymbol{\rho}$. Then,

$$V_{\max} = \max_{\boldsymbol{\rho} \in N_\epsilon} V(\boldsymbol{\pi}_{BR(\boldsymbol{\rho})}, \boldsymbol{\rho})$$

is asymptotically the optimal payoff the learner might receive. We show that the average actual payoff of a NoRA learner will be α-close to this value, in time polynomial in $(1/\alpha, t_\epsilon, \ln(1/\delta), r_{\max}, k \ln k)$. This is a non-intuitive result since the opponents are not assumed to settle to any fixed policy in *finite* time, only in the limit. The theorem says, regardless of the actual policy that the opponents approach asymptotically (which will hence be unknown at all finite times), the learner's average payoff will be close to the best possible *asymptotic* payoff in *polynomial* time.

This result is useful, for instance, against opponents that want to learn some fixed properties of the game through exploration disregarding how its opponents behave, similar to the equilibrium learners Minimax-Q, Nash-Q, CE-Q etc. Here NoRA would efficiently learn the value of the equilibrium, depending on t_ϵ which is tied to the computational complexity of the equilibrium. Therefore, NoRA does not produce a way (per se) to efficiently compute an equilibrium, the complexity of which remains an open problem. Another interesting case is when two no-regret algorithms are guaranteed to converge in policies, e.g., GIGA-WoLF (Bowling 2005) (known to converge against GIGA, also a no-regret algorithm) and ReDVaLeR in self-play (Banerjee & Peng 2004). Our result implies that in such cases, the average actual payoffs of both players (augmented to produce NoRA agents) will be close to the equilibrium payoff *quickly*, assuming that convergence continues to hold with

[3] (Powers & Shoham 2005) use the symbol k to represent the size of the joint action space which is $O(k^n)$ in our notation.

augmentation. Then this result can also be interpreted as showing that the rate of convergence of these algorithms cannot be worse than a polynomial factor of $1/t_\epsilon$.

Theorem 2 *Let $\delta > 0, \alpha > 0$ and $\epsilon < \frac{\alpha}{r_{\max}(1-3\gamma)}$. This implies a t_ϵ. Then there exists a T_0 polynomial in $(1/\alpha, t_\epsilon, \ln(1/\delta), r_{\max}, k\ln k)$ such that if the game is played for at least T_0 rounds, a NoRA agent will achieve an average actual payoff of at least $V_{\max} - \alpha$ against $n-1$ convergent players, with probability at least $1-\delta$.*

Proof : Let $\boldsymbol{\rho}' = \arg\max_{\boldsymbol{\rho}\in N_\epsilon} V(\boldsymbol{\pi}_{BR(\boldsymbol{\rho})}, \boldsymbol{\rho})$. Now we know that $BR_{\boldsymbol{\rho}'}$ contains at least one pure policy, since any action in the support of a best response is also a best response. Let $\boldsymbol{\delta}_{i'} \in BR_{\boldsymbol{\rho}'}$, and $T > t_\epsilon$. The main difference with the previous case of stationary opponents is that $x^T_{i'}$ are not i.i.d. All we know is that

$$E_{j_T}[x^T_{i'} | j_{t_\epsilon+1}, \ldots, j_{T-1}] \geq V_{\max} - r_{\max}\epsilon,$$

since $\rho^T_j \geq \rho'_j - \epsilon$ for all j. But employing the approach of Lemma 5.1 in (Auer *et al.* 1998) (adapted from Neveu) we can show that

$$Z_T = \exp\left(\frac{\lambda'}{r_{\max}} \sum_{t=t_\epsilon+1}^{t=T} (E[x^t_{i'}] - x^t_{i'}) - \frac{\lambda'^2}{r_{\max}} \sum_{t=t_\epsilon+1}^{t=T} E[x^t_{i'}]\right)$$

forms a supermartingale sequence, and that $E[Z_T] \leq 1$. Then by Markov inequality we have $P(Z_T > 2/\delta) \leq \delta/2$. Therefore, with probability at least $1-\delta/2$

$$\sum_{t=t_\epsilon+1}^{t=T} x^t_{i'} \geq (1-\lambda') \sum_{t=t_\epsilon+1}^{t=T} (V_{\max} - r_{\max}\epsilon) - \frac{r_{\max}}{\lambda'} \ln(2/\delta)$$

Combining with equation 5 (where λ is replaced by γ^2/k as in previous section) we get with probability at least $1-\delta$,

$$\begin{aligned} \sum_{t=1}^{t=T} x_{j_t} \quad &\geq \quad (1-2\gamma) \sum_{t=1}^{t=t_\epsilon} x^t_{i'} \\ &\quad +(1-2\gamma)[(1-\lambda')(T-t_\epsilon)(V_{\max} - r_{\max}\epsilon) \\ &\quad -\frac{r_{\max}}{\lambda'} \ln(2/\delta)] - \frac{(1-\gamma) r_{\max} k \ln(2k/\delta)}{\gamma^2} \\ &\quad -(1-\gamma)(p\sqrt{T}+q) \\ &\geq \quad 0 + (1-2\gamma-\lambda')(T-t_\epsilon)(V_{\max} - r_{\max}\epsilon) \\ &\quad -\frac{r_{\max}(1-2\gamma)\ln(2/\delta)}{\lambda'} \\ &\quad -\frac{(1-\gamma) r_{\max} k \ln(2k/\delta)}{\gamma^2} \\ &\quad -(1-\gamma)(p\sqrt{T}+q) \end{aligned}$$

Choosing $\lambda' = \gamma$ and setting the RHS of the average from above, to be at least $V_{\max} - \alpha$ we again end up with a quadratic for minimal $\sqrt{T}$ of the form $ax^2 - bx - c = 0$ where $b, c > 0$ and $a = -r_{\max}\epsilon - 3\gamma V_{\max} + 3\gamma r_{\max}\epsilon + \alpha$. Clearly, $a > 0$ if $\epsilon < \frac{\alpha}{r_{\max}(1-3\gamma)}$ and hence if also $\gamma < 1/3$. Then a real solution exists and the Theorem follows. ■

Note that we do not assume the exact form of dependence of t_ϵ on ϵ. If t_ϵ is a polynomial of $\frac{1}{\epsilon}$ then the Theorem will involve $\frac{1}{\epsilon}$ instead of t_ϵ. However, if t_ϵ happens to be a worse function of $\frac{1}{\epsilon}$ then it will dominate any polynomial in $\frac{1}{\epsilon}$, therefore our guarantee will be in terms of t_ϵ and not $\frac{1}{\epsilon}$.

Since how close the learner gets to $V_{\max}$ cannot be independent of how close the opponents' are to their asymptotic behavior, ϵ will have to be chosen smaller than some function of α. The above theorem allows this function to be no worse than linear. A more technical way of interpreting this dependence goes as follows: since it is not guaranteed that $\lim_{t\to\infty} \boldsymbol{\rho}^t = \boldsymbol{\rho}'$, the asymptotic payoff of the learner may be lower than $V_{\max}$. Hence the uncertainty in its payoff, α, must incorporate the uncertainty in the opponents' asymptotic policy relative to $\boldsymbol{\rho}^t$ which cannot be larger than ϵ for $t > t_\epsilon$.

Arbitrary Opponents

When the opponents can play arbitrary sequence of policies, no-regret learning ensures that the expected average payoff will not be much worse than that of the best response to the opponents' empirical distribution. Though this can be no worse than a minimax [4] or security value of the game, it can possibly be better. We define a variant of the minimax solution that depends on the opponents' policies, called *Opponent Dependent MiniMax* or ODMM. This is produced in the same way as minimax, but allowing minimization over the actual sequence of opponent strategies rather than the entire space of opponent strategies. Therefore, if V_{ODMM} is the expected ODMM payoff and V_{MM} is the expected minimax payoff, then

$$V_{ODMM} \geq V_{MM}.$$

In this section we show that with a high probability a NoRA agent will obtain actual payoffs close to at least V_{ODMM}, in polynomial time. Note that this is a stronger guarantee than Property 3, and is another advantage of our adaptive strategy over meta-level reasoning that can only use the knowledge of the game's fixed minimax solution for security.

Theorem 3 *Let $\epsilon > 3\gamma r_{\max}, \delta > 0$. Then there exists a T_0 polynomial in $(1/\epsilon, \ln(1/\delta), r_{\max}, k\ln k)$ such that if the game is played for at least T_0 rounds, a NoRA agent will achieve an average actual payoff of at least $V_{ODMM} - \epsilon$ against $n-1$ players using arbitrary sequence of policies, with probability at least $1-\delta$.*

Proof : In this case we have even less information about the opponents' policies than the previous two sections. Also, the x^t_i's are not necessarily i.i.d. as in the previous section. Let $\boldsymbol{\eta}$ be any mixture over the learner's action space, A. Using this we can mix equation 5 for all $i = 1, \ldots, k$ (with the previous choice of $\lambda = \gamma^2/k$) to get

$$\begin{aligned} \sum_t x_{j_t} \geq \quad &\sum_t (1-2\gamma)(\boldsymbol{\eta} \cdot \mathbf{x}^t) - \frac{(1-\gamma) r_{\max} k \ln(2k/\delta)}{\gamma^2} \\ &-(1-\gamma)(p\sqrt{T}+q) \end{aligned}$$

[4] Although minimax solutions of two player games are well-defined, it is less concrete for n-player games due to the possibility of collaboration among opponents. We assume all opponents are playing individually without forming coalitions.

since $\sum_i \eta_i = 1$. Then we note that

$$\begin{aligned} E_{j_t}[\boldsymbol{\eta} \cdot \mathbf{x}^t | j_1, \ldots, j_{t-1}] &= \boldsymbol{\eta} \cdot \mathbf{R} \cdot \boldsymbol{\rho}^t \\ &\geq \min_{\{\boldsymbol{\rho}^t\}} \boldsymbol{\eta} \cdot \mathbf{R} \cdot \boldsymbol{\rho}^t \end{aligned}$$

Since this holds for arbitrary $\boldsymbol{\eta}$, it also holds for the $\boldsymbol{\eta}$ that maximizes the RHS above, which means

$$E_{j_t}[\boldsymbol{\eta} \cdot \mathbf{x}^t | j_1, \ldots, j_{t-1}] \geq V_{ODMM}$$

no matter what sequence of policies the opponents play. Then following the same approach of constructing a supermartingale sequence as in the last section, we get the result. ■

Conclusion

We have presented a new class of MARL algorithms, NoRA , by employing a known appraoch of stochastization on a large class of no-regret algorithms. We have shown that its average actual rewards will be close to optimal, with high probability, for two classes of opponents in time polynomial with improved bounds relative to existing results. NoRA also makes minimal assumptions unlike previous approaches. We have also shown that a NoRA agent achieves near optimal average reward, with high probability in polynomial time, against opponents that play converging sequence of policies. The main limitation of NoRA is that it needs to use a low probability of exploration to allow meaningful bounds. It may be given the value of exploration probability with the knowledge of $r_{\max}$ (which then becomes an assumption, still significantly weaker than Assumption 1), and then the values of payoff uncertainties can be lowered to any desired value thus allowing useful bounds. If this knowledge is not available, NoRA may select some small value for γ on its own, without adding to its list of assumptions. But then the guarantees only hold for uncertainty (in payoff) values that are larger than some linear functions of γ, which might make the bounds less useful. One possible future work is to try a decaying schedule for γ instead of a constant. We also intend to test NoRA in repeated games to estimate the values of the constants involved in the polynomial bounds, and compare with Metastrategy.

Acknowledgements

The authors gratefully acknowledge the valuable inputs from anonymous reviewers. This work was supported in part by Louisiana BOR Grant LEQSF(2002-05)-RD-A-29.

References

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 1998. Gambling in a rigged casino: The adversarial multi-armed bandit problem. Technical Report NC2-TR-1998-025, NeuroCOLT2 Technical Report Series.

Banerjee, B., and Peng, J. 2004. Performance bounded reinforcement learning in strategic interactions. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, 2–7. San Jose, CA: AAAI Press.

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

Bowling, M. 2005. Convergence and no-regret in multiagent learning. In *Proceedings of NIPS 2004/5*.

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

Conitzer, V., and Sandholm, T. 2003. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proceedings of the 20th International Conference on Machine Learning*.

Flaxman, A.; Kalai, A.; and McMahan, H. 2005. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. To appear.

Freund, Y., and Schapire, R. E. 1999. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* 29:79 – 103.

Fudenberg, D., and Levine, D. 1995. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control* 19:1065 – 1089.

Greenwald, A., and Hall, K. 2002. Correlated q-learning. In *Proceedings of the AAAI Symposium on Collaborative Learning Agents*.

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

Jafari, A.; Greenwald, A.; Gondek, D.; and Ercal, G. 2001. On no-regret learning, fictitious play, and nash equilibrium. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 226 – 223.

Littlestone, N., and Warmuth, M. 1994. The weighted majority algorithm. *Information and Computation* 108:212 – 261.

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

Littman, M. L. 2001. Friend-or-foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learnig*.

Powers, R., and Shoham, Y. 2005. New criteria and a new algorithm for learning in multi-agent systems. In *Proceedings of NIPS 2004/5*.

Singh, S.; Kearns, M.; and Mansour, Y. 2000. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 541–548.

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

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

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*.