New criteria and a new algorithm for learning in multi-agent systems

Rob Powers
Computer Science Department
Stanford University
Stanford, CA 94305
powers@cs.stanford.edu

Yoav Shoham
Computer Science Department
Stanford University
Stanford, CA 94305
shoham@cs.stanford.edu

Abstract

We propose a new set of criteria for learning algorithms in multi-agent systems, one that is more stringent and (we argue) better justified than previous proposed criteria. Our criteria, which apply most straightforwardly in repeated games with average rewards, consist of three requirements: (a) against a specified class of opponents (this class is a parameter of the criterion) the algorithm yield a payoff that approaches the payoff of the best response, (b) against other opponents the algorithm's payoff at least approach (and possibly exceed) the security level payoff (or maximin value), and (c) subject to these requirements, the algorithm achieve a close to optimal payoff in self-play. We furthermore require that these average payoffs be achieved quickly. We then present a novel algorithm, and show that it meets these new criteria for a particular parameter class, the class of stationary opponents. Finally, we show that the algorithm is effective not only in theory, but also empirically. Using a recently introduced comprehensive game theoretic test suite, we show that the algorithm almost universally outperforms previous learning algorithms.

1 Introduction

There is rapidly growing interest in multi-agent systems, and in particular in learning algorithms for such systems. There is a growing body of algorithms proposed, and some arguments about their relative merits and domains of applicability (for example, [9] and [13]). In [10] we survey much of this literature, and argue that it suffers from not having a clear objective criteria with which to evaluate each algorithm (this shortcoming is not unique to the relatively small computer science literature on multi-agent learning, and is shared by the much vaster literature on learning in game theory). In [10] we also define five different coherent agendas one could adopt, and identify one of them – the agent-centric one – as particularly relevant from the computer science point of view.

In the agent-centric agenda one asks how an agent can learn optimally in the presence of other independent agents, who may also be learning. To make the discussion precise we will concentrate on algorithms for learning in known, fully observable two-player repeated games, with average rewards. We start with the standard definition of a finite stage game (aka. normal form game):

Definition 1 *A two-player stage game is a tuple $G = (A_1, A_2, u_1, u_2)$, where*

- *A_i is a finite set of actions available to player i*
- *$u_i : A_1 \times A_2 \rightarrow \Re$ is a utility function for player i*

Figure 1 shows two well-known games from the literature, to which we'll refer again later.

In a repeated game the stage game is repeated, finitely or infinitely. The agent accumulates rewards at each round; in the finite case the agent's aggregate reward is the average of the stage-game rewards, and in the infinite case it is the limit average (we ignore the subtlety that arises when the limit does not exist, but this case does not present an essential problem).

The strategy space in repeated games is huge, with many strategies naturally viewed as incorporating some sort of learning. For example, in *rational learning* [7], an agent starts with some prior probability on its opponent's strategies, plays the (stage-game) best response, observes the play of the opponents, updates the prior, and repeats. We ask how an agent should learn, bearing in mind that the other agent(s) might also be learning.

While the vast majority of the literature on multi-agent learning (surprisingly) does not start with a precise statement of objectives, there are some exceptions, and we discuss them in the next section, including their shortcomings. In the following section we propose a stronger set of criteria that, we believe, does not suffer from these limitations. We then present an algorithm that provably meets these stronger requirements. However, we believe that all formal requirements – including our own – are merely baseline guarantees, and any proposed algorithm must be subjected to empirical tests. Indeed, previous proposals do provide empirical results and we will show our own results in the last section before our concluding remarks. However, we think it is fair to say that our level of empirical validation is unprecedented in the literature. We not only test all pairwise comparisons of major existing algorithms, but we use a recently-developed game theoretic testbed called GAMUT [8] to systematically sample a very large space of games.

2 Previous criteria for multi-agent learning

To our knowledge, Bowling and Veloso [2] were the first to explicitly put forth formal requirements. Specifically they proposed two criteria:

BV-Property 1 *(Rationality) If the other players' policies converge to stationary policies, then the learning algorithm will converge to a stationary policy that is a best-response (in the stage game) to the other players' policies.*

BV-Property 2 *(Convergence) The learner will necessarily converge to a stationary policy.*

Bowling and Veloso considered known repeated games and proposed an algorithm that provably meets the criteria in 2x2 games (games with two players and two actions per

	$Dare$	$Yield$
$Dare$	$0, 0$	$4, 1$
$Yield$	$1, 4$	$2, 2$

(a) Chicken

	$Cooperate$	$Defect$
$Cooperate$	$3, 3$	$0, 4$
$Defect$	$4, 0$	$1, 1$

(b) Prisoner's Dilemma

Figure 1: Example stage games. The payoff for the row player is given first in each cell, with the payoff for the column player following.

player). Later, Conitzer and Sandholm [4] adopted the same criteria, and demonstrated an algorithm meeting the criteria for all repeated games.

At first glance these criteria are reasonable, but a deeper look is less satisfying. First, note that the property of convergence cannot be applied unconditionally, since one cannot ensure that a learning procedure converges against all possible opponents. So implicit in that requirement is some limitation on the class of opponents. And indeed both [2] and [4] acknowledge this and choose to concentrate on the case of self-play, that is, on opponents that are identical to the agent in question.

We will have more to say about self-play later, but there are other aspects of these criteria that bear discussion. While it is fine to consider opponents playing stationary policies, there are other classes of opponents that might be as relevant or even more relevant; this should be a degree of freedom in the definition of the problem. For instance, one might be interested in the classes of opponents that can be modelled by finite automata with at most k states; these include both stationary and non-stationary strategies.

We find the property of requiring convergence to a stationary strategy particularly hard to justify. Consider the Prisoner's Dilemma game in Figure 1. The Tit-for-Tat algorithm[1] achieves an average payoff of 3 in self-play, while the unique Nash equilibrium of the stage game has a payoff of only 1. Similarly, in the game of Chicken, also shown in Figure 1, a strategy that alternates daring while its opponent yields and yielding while its opponent dares achieves a higher expected payoff than any stationary policy could guarantee in self-play. This brings up a fundamental point; both of the existing proposals can be thought of as a requirement on the *play* of the agent, rather than the *reward* the agent receives.

Our final point regarding these two criteria is that they express properties that hold in the limit, with no requirements whatsoever on the algorithm's performance in any finite period.

3 A new set of criteria for learning

In our approach we wish to keep the notion of optimality against a specific set of opponents. But instead of restricting this set in advance, we'll make this a parameter of the properties. Acknowledging that we may encounter opponents outside our target set, we also want a bound on our payoff in that situation. We propose a security value, $V_{Security}$, defined as $\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2} EV(\pi_1, \pi_2)$.[2] This is equivalent to the game-theoretic notion of a maximin value for a stage game. As a possible motivation for our approach, consider the game of Rock Paper Scissors, which despite its simplicity has motivated several international tournaments. While the unique game-theoretically "optimal" policy is to randomize, the winners of the tournaments are those players who can most effectively exploit their opponents without being exploited in turn.

The question remains of how best to handle self-play. One method would be to require that our algorithm be added to the set of opponents it is required to play a best response to. While this may seem appealing at first glance, it can be a very weak requirement on the actual payoff the agent receives. Since our opponent is no longer independent of our choice of strategy, we can do better than settling for just any mutual best response, and try to maximize the value we achieve as well. We therefore propose requiring the algorithm achieve at least the value of some Nash equilibrium that is Pareto efficient over the set of

[1] The Tit-for-Tat algorithm cooperates in the first round and then for each successive round plays the action its opponent played in the previous round.

[2] Throughout the paper, we will use $EV(\pi_1, \pi_2)$ to indicate the expected payoff to a player for playing strategy π_1 against an opponent playing π_2 and $EOV(\pi_1, \pi_2)$ as the expected payoff the opponent achieves. Π_1 and Π_2 are the sets of mixed strategies for the agent and its opponent respectively.

Nash equilibria.[3]

In each of the following properties, let k be the number of outcomes for the game and b the maximum possible difference in payoffs across the outcomes.

PS-Property 1 *For any choice of $\epsilon > 0$ and $\delta > 0$ there should exist a T_0, polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, k$, and b, such that for any number of rounds $t > T_0$ the algorithm achieves average payoff of at least $V_{BR} - \epsilon$ against a member of the selected set of opponents with probability $1 - \delta$, where V_{BR} is the value of the best response to the actual opponent.*

PS-Property 2 *For any choice of $\epsilon > 0$ and $\delta > 0$ there should exist a T_0, polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, k$, and b, such that for any number of rounds $t > T_0$ the algorithm achieves average payoff of at least $V_{selfPlay} - \epsilon$ in self-play with probability $1-\delta$, where $V_{selfPlay}$ is defined as the minimum value achieved by any Nash equilibrium that is not Pareto dominated by another Nash equilibrium.*

PS-Property 3 *For any choice of $\epsilon > 0$ and $\delta > 0$ there should exist a T_0, polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}, k$, and b, such that for any number of rounds $t > T_0$ the algorithm achieves average payoff of at least $V_{security} - \epsilon$ against any opponent with probability $1-\delta$, where $V_{security}$ is the agent's security value for the stage game.*

4 An algorithm

We can now use the above criteria as guidelines for devising a new algorithm for the class of stationary opponents. Our method incorporates modifications of three simple strategies: Fictitious Play [1], Bully [13], and the maximin strategy in order to create a more powerful hybrid algorithm.

Fictitious Play has been shown to achieve the best response against a stationary opponent in the limit. Each round it plays its best response to the most likely stationary opponent given the prior history of play. In our implementation we use a somewhat more generous best-response calculation in order to achieve our performance requirements during self-play.

$$BR_\epsilon(\pi) \leftarrow \arg\max_{x \in X}(EOV(x, \pi)),^4$$
$$\text{where } X = \{y \in \Pi_1 : EV(y, \pi) \geq \max_{z \in \Pi_1}(EV(z, \pi)) - \epsilon\}$$

We extend the Bully algorithm to consider the full set of mixed strategies and again maximize our opponent's value when multiple strategies yield equal payoff for our agent.

$$BullyMixed \leftarrow \arg\max_{x \in X}(EOV(x, BR(x))),$$
$$\text{where } X = \{y \in \Pi_1 : EV(y, BR_0(y)) = \max_{z \in \Pi_1}(EV(z, BR_0(z)))\}$$

The maximin strategy is defined as

$$Maximin \leftarrow \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2} EV(\pi_1, \pi_2)$$

We will now show how to combine these strategies into a single method satisfying all three criteria. In the code shown below, t is the current round, $AvgValue_m$ is the average value achieved by the agent during the last m rounds, V_{Bully} is shorthand for

[3] An outcome is Pareto efficient over a set if there is no other outcome in that set with a payoff at least as high for every agent and strictly higher for at least one agent.

[4] Note that $BR_0(\pi)$ is a member of the standard set of best response strategies to π.

$EV(BullyMixed, BR_0(BullyMixed))$, and $d_{t_1}^{t_2}$ represents the distribution of opponent actions for the period from round t_1 to round t_2.

```
Set strategy = BullyMixed
for τ1 time steps
  Play strategy
for τ2 time steps
  if (strategy == BullyMixed AND AvgValue_H < V_Bully − ε1)
    With probability, p, set strategy = BR_ε2(d_0^t)
  Play strategy
if ||d_0^τ1 − d_{t−τ1}^t|| < ε3
  Set bestStrategy = BR_ε2(d_0^t)
else if (strategy == BullyMixed AND AvgValue_H > V_Bully − ε1)
  Set bestStrategy = BullyMixed
else
  Set bestStrategy = BestResponse
while not end of game
  if avgValue_{t−τ0} < V_security − ε0
    Play maximin strategy for τ3 time steps
  else
    Play bestStrategy for τ3 time steps
```

The algorithm starts out with a coordination/exploration period in which it attempts to determine what class its opponent is in. At the end of this period it chooses one of three strategies for the rest of the game. If it determines its opponent may be stationary it settles on a best response to the history up until that point. Otherwise, if the BullyMixed strategy has been performing well it maintains it. If neither of these conditions holds, it adopts a default strategy, which we have set to be the BestResponse strategy. This strategy changes each round, playing the best response to the maximum likelihood opponent strategy based on the last H rounds of play. Once one of these strategies has been selected, the algorithm plays according to it whenever the average value meets or exceeds the security level, reverting to the maximin strategy if the value drops too low.

Theorem 1 *Our algorithm satisfies the three properties stated in section 3 for the class of stationary opponents, with a T_0 proportional to $(\frac{b}{\epsilon})^3 \frac{1}{\delta}$.*

While this theorem can be proven for all three properties using a lengthy combination of basic probability theory and repeated applications of the Hoeffding inequality [6], let's take a brief look at how the algorithm satisfies the self-play criteria. Initially both agents will be playing BullyMixed. There are two cases we need to handle, depending on what payoff the agents achieve when both are playing BullyMixed. Let's consider the case where for at least one player this payoff is less than $V_{Bully} - \epsilon$. We can see that for a large enough value of τ_2 it becomes highly likely that one of the algorithms will change to a best response during τ_2. After H more rounds the $AvgValue_H$ for the other player will become quite close to V_{Bully} and that player will become unlikely to switch away from BullyMixed during the rest of τ_2. By choosing a small enough value of p we can minimize the chance that the second player will also switch during the length H period of realignment. If we get to the end of the initial phase with just one agent playing BullyMixed, let's call it agent 1, we know agent 1 will not think its opponent is stationary, since if BullyMixed_2 was close to $BR(BullyMixed_1)$ then $EV(BullyMixed_2, BullyMixed_1)$ would be close to V_{Bully}. At the same time, agent 2 will have observed a stationary opponent so will adopt a best-response. Given our definitions of the best-response function and BullyMixed we can then show that the expected value for each agent is at least $V_{selfPlay}$.

5 Empirical results

While the ability to prove that our algorithm satisfies the criteria we put forth is comforting, we feel this is but a first step in making a compelling argument that an approach might be useful in practice. Traditionally, researchers putting forth a new algorithm have also included an empirical comparison of that algorithm with previous work. While we think this is a critical component of evaluating an algorithm, most prior work has tested their algorithm against just one or two other algorithms on a very narrow set of test environments, which often vary from researcher to researcher. This practice has made it hard to compare the performance of different algorithms in a consistent fashion.

In order to address this situation, we've started to code a collection of existing algorithms. Combining this set of algorithms with a wide variety of repeated games from GAMUT [8], a game theoretic test suite, we have the beginnings of a comprehensive testbed for multi-agent learning algorithms. In the rest of this section, we'll concentrate on the results for our algorithm, but we hope that this testbed can form the foundation for a broad, consistent framework of empirical testing in multi-agent learning going forward.

For all of our environments we conducted our tests using a tournament format, where each algorithm plays all other algorithms including itself.

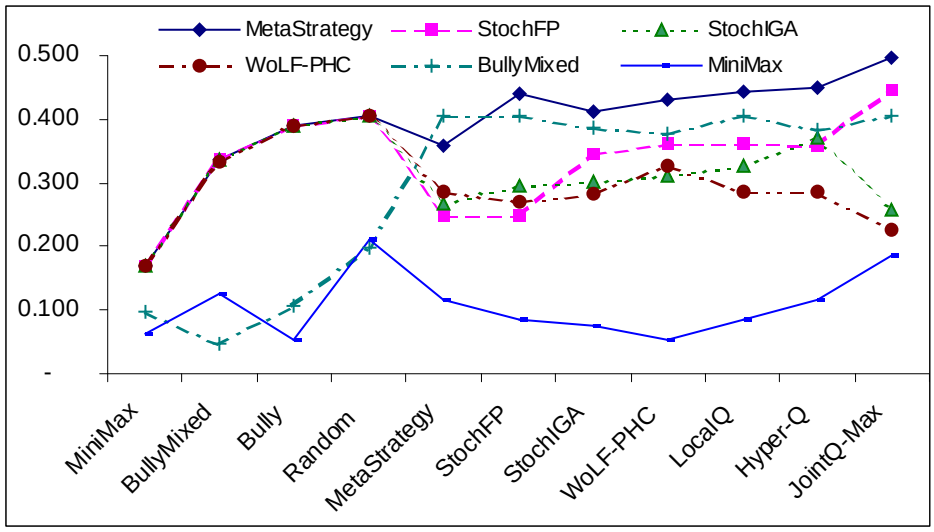

Figure 2: Average value for last 20K rounds (of 200K) across all games in GAMUT.

Let us first consider the results of the tournament over a full set of games in GAMUT. Figure 2 portrays the average value achieved by each agent (y-axis) averaged over all games, when playing each possible opponent (x-axis). The set of agents includes our strategy (MetaStrategy), six different adaptive learning approaches (Stochastic Fictitious Play [1,5], Stochastic IGA[11], WoLF-PHC[2], Hyper-Q learning[14], Local Q-learning [15], and JAL-Max [3] (which learns Q-values over the joint action space but assumes its opponent will cooperate to maximize its payoff)), and four fixed strategies (BullyMixed, Bully [12], the maximin strategy, and Random (which selects a stationary mixed strategy at random)). We have chosen a subset of the most successful algorithms to display on the graph. We can see that against the four stationary opponents, all of the best responders fared equally well, while the fixed strategy players achieved poor rewards. In contrast, BullyMixed fared quite well against the best-responding algorithms. As desired, our new algorithm was able to combine the best of these characteristics to achieve the highest value

against all opponents except itself. The reason it fares worse than BullyMixed when playing against itself is that it will always yield to BullyMixed, giving away the more advantageous outcome. However, when comparing how each agent performs in self-play, our algorithm scores quite well, finishing a close second to Hyper-Q learning while the two Bully algorithms finish near last. Hyper-Q is able to gain in self-play by occasionally converging to outcomes with high social welfare that our strategy does not consider.

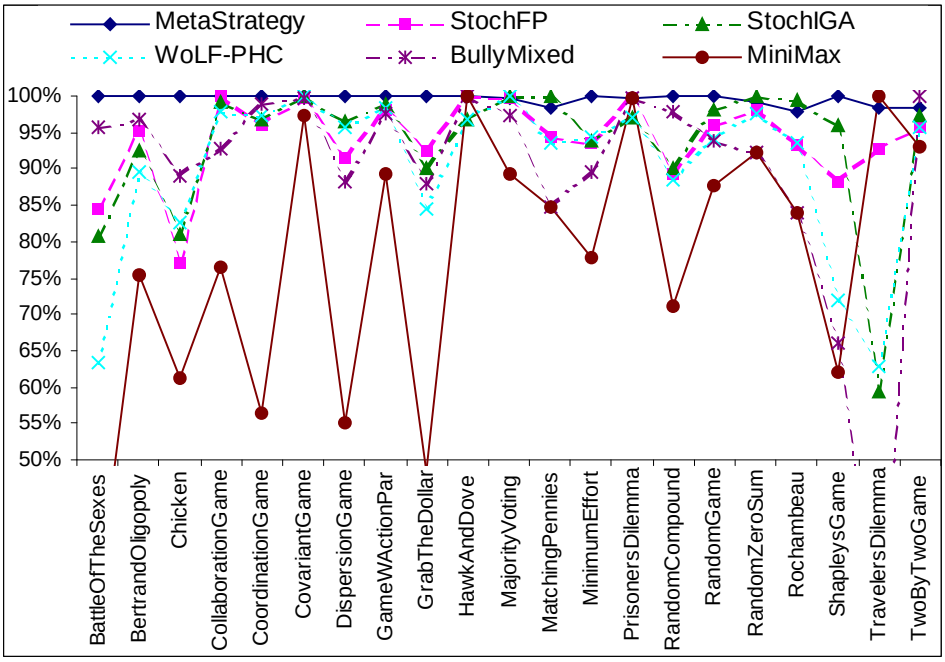

Figure 3: Percent of maximum value for last 20K rounds (of 200K) averaged across all opponents for selected games in GAMUT. The rewards were divided by the maximum reward achieved by any agent to make visual comparisons easier.

So far we've seen that our new algorithm performs well when playing against a variety of opponents. In Figure 3 we show the reward for each agent, averaged across the set of possible opponents for a selection of games in GAMUT (we omitted games where the algorithms achieved virtually identical payoffs). Once again our algorithm outperforms the existing algorithms in nearly all games. When it fails to achieve the highest reward it often appears to be due to its policy of "generosity"; in games where it has multiple actions yielding equal value, it chooses a best response that maximizes its opponent's value.

The ability to study how individual strategies fare in each class of environment reflects an advantage of our more comprehensive testing approach. In future work, we can use this data both to aid our selection of an appropriate algorithm for a new environment and to pinpoint areas where our algorithm might be improved. Note that we use environment here to indicate a combination of both the game and the distribution over opponents.

6 Conclusions and Future Work

Our objective in this work was to put forth a new set of criteria for evaluating the performance of multi-agent learning algorithms as well as propose a more comprehensive method for empirical testing. In order to motivate this new approach for vetting algorithms, we have

presented a novel algorithm that meets our criteria and outperforms existing algorithms in a wide variety of environments. We are continuing to work actively to extend our approach. In particular, we wish to demonstrate the generality of our approach by providing algorithms that calculate best response to different sets of opponents (conditional strategies, finite automata, etc.) Additionally, the criteria need to be generalized for n-player games and we hope to combine our method for known games with methods for learning the structure of the game, ultimately devising a new algorithm for unknown stochastic games.

Acknowledgements

This work was supported in part by a Benchmark Stanford Graduate Fellowship, DARPA grant F30602-00-2-0598, and NSF grant IIS-0205633.

References

[1] Brown, G. (1951). Iterative Solution of Games by Fictitious Play. In *Activity Analysis of Production and Allocation*. New York: John Wiley and Sons.

[2] Bowling, M. & Veloso, M. (2002). Multiagent learning using a variable learning rate. In *Artificial Intelligence, 136*, pp. 215-250.

[3] Claus, C. & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the National Conference on Artificial Intelligence* , pp. 746-752.

[4] Conitzer, V. & 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*, pp. 83-90, Washington, DC.

[5] Fudenberg, D. & Levine, D. (1998). *The theory of learning in games*. MIT Press.

[6] Hoeffding, W. (1956). On the distribution of the number of successes in independent trials. *Annals of Mathematical Statistics* 27:713-721.

[7] Kalai, E. & Lehrer, E. (1993). Rational learning leads to Nash equilibrium. In *Econometrica*, 61(5):1019-1045.

[8] Nudelman, E., Wortman, J., Leyton-Brown, K., & Shoham, Y. (2004). Run the GAMUT: A Comprehensive Approach to Evaluating Game-Theoretic Algorithms. *AAMAS-2004*. To Appear.

[9] Sen, S. & Weiss, G. (1998). Learning in multiagent systems. In *Multiagent systems: A modern introduction to distributed artificial intelligence*, chapter 6, pp. 259-298, MIT Press.

[10] Shoham, Y., Powers, R., & Grenager, T. (2003). Multi-Agent Reinforcement Learning: a critical survey. Technical Report.

[11] Singh, S., Kearns, M., & Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. In *Proceedings of UAI-2000*, pp. 541-548, Morgan Kaufman.

[12] Stone, P. & Littman, M. (2001). Implicit Negotiation in Repeated Games. In *Pre-proceedings of the Eighth International Workshop on Agent Theories, Architectures, and Languages*, pp. 96-105.

[13] Stone, P. & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3).

[14] Tesauro, G. (2004). Extending Q-Learning to General Adaptive Multi-Agent Systems. In *Advances in Neural Information Processing Systems 16*.

[15] Watkins, C. & Dayan, P. (1992). Technical note: Q-learning. *Machine Learning*, 8(3/4):279-292.