Learning in Real-Time in Repeated Games Using Experts (Extended Abstract)

Jacob W. Crandall
Masdar Institute of Science and Technology
Abu Dhabi, UAE
jcrandall@masdar.ac.ae

ABSTRACT

Despite much progress, state-of-the-art learning algorithms for repeated games still often require thousands of moves to learn effectively – even in simple games. Our goal is to find algorithms that learn to play effective strategies in tens of moves in many games when paired against various associates. Toward this end, we describe a new meta-algorithm designed to increase the learning speed and proficiency of expert algorithms. We show that this meta-algorithm enhances four expert algorithms so that they quickly learn effective strategies in two-player repeated games.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Multi-agent learning, game theory, experts, regret

1. INTRODUCTION

Learning to adapt to other agents during repeated interactions has been well-studied over the past several decades. Given sufficient time, existing algorithms are often capable of learning effective strategies in repeated games. Nevertheless, state-of-the-art learning algorithms typically learn rather slowly in these games. They often require thousands of moves to learn effective strategies even in simple games. Algorithms that do learn quickly often produce myopic solutions that yield low payoffs.

The inability to quickly learn effective strategies has prohibited multi-agent learning algorithms from being used in real-world systems in which devices and people repeatedly interact. Such applications include power systems, computer networks, social networks, and electronic commerce. Thus, our goal is to find algorithms for repeated general-sum games that learn to play effective strategies when associating with both static and learning agents within tens of moves.

Expert algorithms have the potential to achieve this goal. In each time step, an expert algorithm selects an expert from a set of experts to dictate its behavior. Despite their potential, we show that existing expert algorithms do not quickly learn to select an effective expert in many repeated games played against other learning algorithms. We then overview a meta-algorithm that can be applied to these expert algorithms to substantially improve their effectiveness both in terms of learning speed and average payoffs.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

2. EVALUATING EXPERT ALGORITHMS

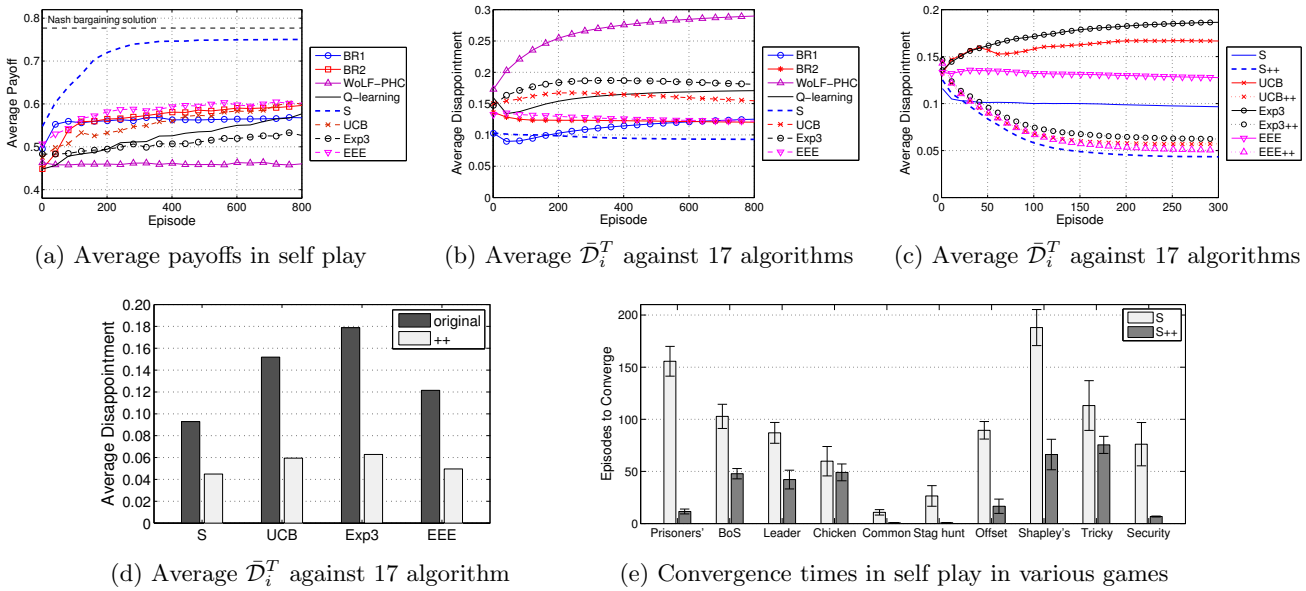
We consider two-player repeated normal-form games consisting of a set of joint actions $A = A_1 \times A_2$, where A_i is player (or agent) i's action set, and a payoff function $M : A \rightarrow \mathbb{R}^2$. In each episode (or time) t, each agent i independently selects an action $a_i^t \in A_i$. The resulting joint action $\mathbf{a}^t = (a_1^t, a_2^t)$ produces the payoff profile $\mathbf{r}^t = (r_1^t, r_2^t)$, where r_i^t is the payoff to agent i. Play repeats an unknown number of episodes. We refer to the two agents as i and $-i$. Also, let Φ_i denote agent i's set of experts.

Regret is commonly used to evaluate expert algorithms. Loosely, regret refers to the difference between the payoffs agent i received and what it would have received had it always followed its best expert and *had its associate's actions remained unchanged*. Since this latter assumption is often false in repeated games [4], regret often does not correlate with the agent's actual payoffs. Thus, we propose and use a new metric, called *disappointment*, for measuring the success of an expert algorithm in repeated games. Disappointment is similar to regret, but does not assume that agent i's actions do not impact its associate's actions. Formally, agent i's *total disappointment* up to time T is

$$\mathcal{D}_i^T = \max_{\phi_i \in \Phi_i} \sum_{t=1}^{T} \left(u_i^t(\phi_i, \pi_{-i}^t(\phi_i)) - r_i^t \right), \quad (1)$$

where $\pi_{-i}^t(\phi_i)$ is the strategy that agent $-i$ would have played in episode t had agent i always followed expert ϕ_i up to episode t, and $u_i^t(\phi_i, \pi_{-i}^t(\phi_i))$ is agent i's expected payoff in episode t if it had always followed expert ϕ_i and agent $-i$ had played $\pi_{-i}^t(\phi_i)$ in each episode t. Agent i's *average disappointment* up to episode T is $\bar{\mathcal{D}}_i^T = \mathcal{D}_i^T / T$.

Unlike regret, disappointment is directly connected to accumulated payoffs. Algorithms that receive higher payoffs against a given associate achieve lower disappointment than algorithms that receive lower payoffs against that associate. While it is impossible for an algorithm to be guaranteed to have no disappointment against an arbitrary associate without being omniscient, we seek to find expert algorithms that *quickly* learn effective strategies (i.e., low disappointment) in many games played against many algorithms.



(a) Average payoffs in self play

(b) Average $\bar{\mathcal{D}}_i^T$ against 17 algorithms

(c) Average $\bar{\mathcal{D}}_i^T$ against 17 algorithms

(d) Average $\bar{\mathcal{D}}_i^T$ against 17 algorithm

(e) Convergence times in self play in various games

Figure 1: Results averaged over the ten selected repeated matrix games shown on the x-axis of (e).

3. RESULTS – EXISTING ALGORITHMS

We first evaluate the performance of four existing expert algorithms, namely Exp3 [2], UCB1 [1], EEE [4], and S [5]. We provide these algorithms with a set of experts consisting of multiple leader automata patterned after [3] and multiple forms of best-response automata. We then paired these algorithms against 17 different algorithms, including themselves and other (learning and static) algorithms, in the ten matrix games shown on the x-axis of Figure 1(e). Figure 1(a) shows the average per-episode payoffs of the agents in self play. The figure shows that these and other learning algorithms do not learn effectively within tens of moves, and (with the exception of S) their average payoffs fall far short of the Nash bargaining solution over all episodes. Furthermore, these algorithms have high disappointment in these games outside of self play (Figure 1(b)).

4. A NEW ALGORITHM

Given the inability of these expert algorithms to quickly learn effective strategies in these games, we propose a new meta-algorithm to improve their performance. This meta-algorithm computes a reduced set of experts $\Phi_i^{\text{reduced}} \subseteq \Phi_i$ which consists only of those experts that could potentially produce payoffs that meet or exceed agent i's aspiration level α_i^t. The highest potential payoff of each expert $\phi_i \in \Phi_i$ is estimated by reasoning over multiple opponent models. The aspiration level α_i^t is determined using aspiration learning [5] after initially setting α_i^0 to the highest potential among all experts $\phi_i \in \Phi_i$. The reduced set Φ_i^{reduced} is provided to the expert algorithm in each episode in place of Φ_i.

5. RESULTS – NEW ALGORITHM

We used this meta-algorithm to form four new algorithms: Exp3++, UCB++, EEE++, and S++. The average disappointments of these algorithms in the same selected games are shown in Figures 1(c) and 1(d). The meta-algorithm substantially improves each of the four algorithms both in early and later episodes. The enhanced algorithms quickly achieve low disappointment on average. These results also hold for individual games and pairings (not shown). Coupled with Figure 1(e), these results show that S++, the best performing of these algorithms, learns effective strategies within tens of moves in these games.

6. CONCLUSION

In this short paper, we have proposed a new metric (called disappointment) for evaluating expert algorithms in repeated games. We also overviewed a new meta-algorithm designed to enhance expert algorithms so that they quickly learn more effective strategies when paired against various associates in many repeated games. This meta-algorithm could also be applied in more complex games (such as stochastic games) given sophisticated experts designed for these games.

7. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proc. of the 36th Symp. on the Foundations of CS*, pages 322–331, 1995.

[3] J. W. Crandall and M. A. Goodrich. Learning to teach and follow in repeated games. In *AAAI workshop on Multiagent Learning*, 2005.

[4] D. de Farias and N. Megiddo. Exploration–exploitation tradeoffs for expert algorithms in reactive environments. In *Advances in Neural Information Processing Systems 17*, pages 409–416, 2005.

[5] R. Karandikar, D. Mookherjee, D. R., and F. Vega-Redondo. Evolving aspirations and cooperation. *Journal of Economic Theory*, 80:292–331, 1998.