Fast Learning against Adaptive Adversarial Opponents

Mohamed Elidrisi, Nicholas Johnson, and Maria Gini
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
{elidrisi,njohnson,gini}@cs.umn.edu

ABSTRACT

The ability to learn and adapt when playing against an adaptive opponent requires the ability to predict the opponent's behavior. Capturing any changes in the opponent's behavior during a sequence of plays is critical to achieve positive outcomes in such an environment.

We identify two new requirements that we suggest are essential for agents that learn in adaptive environments. These requirements are dictated by the fact that repeated interactions in practice have to be limited and that the opponent can rapidly change strategy through the sequence of interactions. We believe that building intelligent agents that can survive in environments with such requirements will lead to wider deployment of learning agents.

We propose a novel algorithm that is able to learn and adapt rapidly to an opponent even when the number of interactions is limited and the opponent is adapting quickly by changing its strategy. The context we use for the experimental work is two player normal form games. We compare the performance of an agent using our algorithm against agents using existing multiagent learning algorithms.

1. INTRODUCTION

Learning in the presence of opponents is challenging not only because the opponent's behavior is unknown, but also because the opponent can change behavior and adapt to other players. We assume the opponent is adversarial in nature and that the learning agent does not know what strategy the opponent plays. Modeling the behavior of the opponent is a key to success. Moreover, capturing any changes in the opponent's behavior could help avoid issues of deception and exploitation by the opponent.

Multiagent learning has emerged as an active area of research in the past decade with algorithms that can function in adaptive environments as well as theoretical properties for new algorithms [14]. However, the deployment of such algorithms to real world application remains limited due to some unrealistic constraints. In this work we identify two constraints that have appeared in previous work [9, 16] that we believe could potentially increase the deployment of multiagent learning algorithms.

One of the major constraints initially set for agents that play against an opponent in a competitive environment is the assumption that the opponent is either stationary or will converge to a stationary policy [4]. New criteria that, to some degree, relax the stationarity assumption have been proposed in [14]. However, there are still some critical assumptions that we believe are obstacles for the use of learning agents in real world domains.

The first obstacle relates to the need for extremely long sequences of interactions between the agents, often in the order of hundreds of thousands, before the agent learns how to play against the opponent. The second relates to the fact that abrupt changes in the opponent's policy often require restarting the learning process. The second issue has been analyzed from different view points in terms of data bias and computing best-response in games [11].

We proposed an algorithm for fast learning (FAL) against an opponent that deals with those constraints [9]. In this paper we briefly describe FAL and introduce a more refined algorithm, called *FAL-Ensemble*, that is capable of detecting changes in the opponent's behavior faster than the original FAL. We show experimentally how the algorithm performs compared to other commonly used multiagent learning algorithms in scenarios that have the following properties:

1. **Limited repeated interactions.** We consider cases where agents have a limited number of repeated interactions with an adaptive opponent and do not have a long term history to build an opponent model. An intelligent agent deployed in the world is typically faced with a task and a limitation in its possible repeated interactions with its opponent. This is often true for the general case of any opponent but is even more relevant when the opponent is a human. Repeated interactions beyond a few hundred games are a luxury in the world and a rare commodity when an actual human is at the other end. Current literature typically shows empirical results in thousands of interactions even for simple normal form games. While convergence to equilibrium in the long term is very important theoretically, it doesn't address the issue of how to deal with a limited number of interactions.
2. **Rapidly adaptive opponents.** An abrupt policy change requires an agent to be able to learn fast so it can adapt to the changes in the opponent's play. There are numerous reasons for an opponent to change its policy such as the ability to deceive its opponents. The opponent could initially adopt a suboptimal strategy to confuse the learning agent and then switch to another strategy that could lead to more gains.

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

2. RELATED WORK

Multiagent learning has been an active area of research in the past decade, with the emergence of many criteria and theoretical foundations for developing new algorithms. Within the AI community, the problem has been addressed mainly from three different approaches, as elaborated in the extensive survey by Busoniu et al. [6].

The first approach is focused on adapting single agent reinforcement algorithms for the multiagent setting, as in Littman [12] and other multiagent reinforcement learning methods, but incorporating the adversarial nature of the environment. The second approach combines policy search methods with knowledge of the adversarial nature, as in the work of Bowling [4]. The third approach starts initially from a game theoretic perspective and attempts to produce an algorithmic framework as in [13, 15] and others.

We describe briefly a few algorithms from the different approaches. Minimax-Q [12] was one of the early algorithms with a clear framework for the multiagent learning problem. Minimax-Q is a modified Q-Learning algorithm that uses minimax instead of max in the action choosing step. The performance of Minimax-Q is demonstrated in a two-player grid-like game of soccer.

Godfather and Bully were introduced in [13] as examples of leader strategies that can influence the best response of follower agents in repeated games. Such strategies were shown to be advantageous in some normal form games. They represent a class of deterministic strategies that act with full knowledge of the game structure and the opponent's previous actions. Bully plays the minimax strategy that obtains the security value, which is the minimum value a player can obtain regardless of what the opponent does. Godfather is a generalization of Tit-For-Tat [13] that offers the opponent the opportunity to choose an action with a reward higher than its security value. If the opponent refuses the offer Godfather punishes it.

The "Win or Learn Fast"(WOLF) [4] principle can be applied in any setting where an algorithm is learning with some learning rate. The idea is that as long as the agent is winning the learning rate is increased by a variable α but when losing another rate $\beta > \alpha$ is used. Empirical results running a policy hill climbing (WOLF-PHC) algorithm show the algorithm will converge to the optimal policy when playing against a stationary opponent. Note that if the learning rate used is 1 then WOLF-PHC is equivalent to Q-Learning.

Bowling and Veloso proposed two criteria for multiagent learning, rationality and convergence. *Rationality* states that if the opponents converge to stationary policies then the learning algorithm will converge to a best response. *Convergence* states that the learner will converge to a stationary policy.

Powers and Shoham [15] extended those criteria to include a larger class of opponents. They parameterize the class of opponents against which they achieve optimal performance. This expands the set of allowable opponents from just opponents with stationary policies to adaptive opponents. However, to guarantee the algorithmic properties, the length of history the opponent can use is limited to only the last k moves. Without this assumption an agent will see the history only once and will be unable to learn. The class of adaptive opponents is restricted to a target class known *a priori* and to learn the agent needs to play a very long training sequence. In the results presented the training sequence is 200K rounds for small matrix games. Their algorithm guarantees the following: (1) *Targeted Optimality*: the agent's payoff approaches best response after a phase of exploration against the target class of opponents. (2) *Optimality*: the agent is Pareto efficient in self-play meaning that it cannot be dominated. (3) *Security* : the payoff against any other opponent approaches the security value.

Adapt When Everybody is Stationary, Otherwise Move to equilibrium (AWESOME) [8] is an algorithm for learning against stationary opponents by observing the opponent's actions. AWESOME precomputes a Nash equilibrium for the game and at each iteration of the game it checks whether the opponent is playing the precomputed Nash equilibrium or is playing a non-stationary strategy. It performs the check by defining an epoch and comparing the distribution of actions in the current epoch to the precomputed distribution of actions for the Nash equilibrium. If it determines that the opponent is not playing the Nash equilibrium it then compares the distribution of actions in the current epoch to the previous epoch to determine whether the opponent is playing some other stationary strategy. If it is then AWESOME plays its best response otherwise it restarts. After each restart AWESOME forgets what it has learned and begins by playing the Nash equilibrium and performing the strategy checks. AWESOME learns the best response against a stationary opponent and in self-play.

ReDValer [2] also works against stationary players and observes the opponent's mixed strategy. ReDValer has an additional feature that guarantees a constant regret. The algorithm NoRa [3] is a follow up on ReDVaLeR which modifies existing no-regret algorithms for the multiagent setting. The proposed no-regret learning algorithm satisfies a modified version of Targeted Optimality and Security. The algorithm provides guarantees against stationary opponents, opponents converging to a stationary policy, and produces an average expected payoff not much worse than what best response to the observed distribution from the opponent would produce. A limitation of no-regret strategies is that they do not capitalize on patterns in the opponent play [15] and do not account for an opponent whose strategy depends on the agent's moves.

Weighted Policy Learner (WPL) [1] is an algorithm that converges to a Nash equilibrium with limited knowledge (i.e. no knowledge of the underlying game or observation of the opponent's action). The algorithm works in two-player two-action games with limited knowledge where other multiagent learning algorithms fail.

Efficient Learning Equilibrium (ELE) [5] handles scenarios with imperfect information games. However, it requires that the learning algorithm itself be in an equilibrium. It computes the surplus the agent would have from playing the Nash equilibrium and shows that deviations from such learning algorithm are irrational in polynomial time.

Learn or Exploit in Adversary Induced Markov decision process (LoE-AIM) [7] is an algorithm that has target optimality against any opponent with bounded memory. The memory bound is important because without memory bound the Markov decision process induced by the adversary joint histories will have an infinite state space. The algorithm is distinctive because in theory it makes no assumption about a target class. However, in the implementation it assumes it knows if the opponent is stationary or not.

3. FAST ADAPTIVE LEARNER (FAL)

We proposed a novel algorithm for an agent which learns a strategy to use when playing repeated games against an adaptive opponent [9]. The key feature of our algorithm is that it is able to learn in a limited number of repeated interactions and is able to detect and adapt to potentially abrupt changes in the opponent's strategy.

The algorithm, at a high level, has two parts: (1) a *Predictive Model* which makes a prediction about the opponent's next action; (2) a *Reasoning Model* which chooses a suitable best response accordingly.

There is a large class of models and methods that can be used for both parts of the algorithm. However, in order to meet the requirements of limited repeated interactions and fast adaptive opponents we need to impose constraints on the models. These are the main requirements on each part:

1. the *Predictive model* makes a prediction about the opponent's next action with the following requirements:
   (a) is online in nature;
   (b) weighs the recent observation more than the past;
   (c) preserves the ordering of the sequence of repeated interactions;
   (d) is sensitive to abrupt changes in data streams.
2. The *Reasoning model* chooses a suitable best response with the following requirements:
   (a) reasons about the temporal accuracy of the predictive model above;
   (b) reasons if the opponent is cooperative or competitive;
   (c) attempts to teach the opponent to cooperate (if the opponent is teachable);
   (d) analyzes its own average reward and whether it is "losing" or "winning" and incorporates it in its decision making process.

We instantiate our *Fast Adaptive Learner (FAL)* algorithm with the models we describe next.

3.1 Predictive Model

For the Predictive Model, we use an algorithm called Entropy Learning Pruned Hypothesis Space (ELPH) [10]. ELPH was developed as an online predictive algorithm for sequences that met the requirements we specified above for the predictive model. ELPH has the ability to learn to predict from a sequence of observations rapidly and therefore can adapt to non-stationary opponents quickly.

At the core of ELPH is the hypothesis space, which is constantly updated with patterns and predictions using the recent history of events and predictions. Every time an observation matches one or more patterns, ELPH computes the entropy of the matching set and uses the entropy to prune the rules that have high entropy. The rule with the lowest entropy is then used to make the prediction. Pruning facilitates adaptation because it removes unsuccessful rules from the hypothesis space. Algorithm 1 provides a sketch of how ELPH works, more details are in [10].

3.2 Reasoning Model

For the Reasoning Model, we adapt a solution which is a compromise of two strategies: (1) a Godfather like strategy, called *Godfather-Future*, which takes as input the predicted future opponent action; (2) a model, called *Meta-Prediction*, which reasons about the success of the predictive model in predicting the opponent's next action.

Algorithm 1 ELPH: Entropy Learning Pruned Hypothesis Space

```
λ: is the threshold for hypothesis pruning.
{a_1, a_2, ..., a_k} is the history of k actions.
create all possible subsets of the history {a_1, a_2, ..., a_k}
compute the reliable entropy for each subset
prune subsets with entropy greater than λ
make a prediction based on the lowest entropy hypothesis.
```

The Godfather-Future strategy computes a targetable pair of actions, i.e. any pair of deterministic strategies with the property that it yields a reward for the players higher than their security value.

While the original Godfather plays its half of the targetable pair if the opponent played its half in the last interaction, Godfather-Future plays its half of the targetable pair if the opponent is predicted to play its half in the next interaction.

The Meta-Prediction computes a weighted average prediction success on a moving horizon. Let's assume that $P_i \in \{0, 1\}$ is the prediction of the predictive model in interaction i, $P_i = 1$ if it predicted successfully and 0 otherwise. The weighted W_t average prediction success at time t is

$$W_t = \frac{(t)P_t + (t-1)P_{t-1} + \ldots + (t-k)P_{t-k}}{t + (t-1) + \ldots + (t-k)}$$

where k is a constant.

Meta-Prediction compares the value of W_t to a parameter β. If $W_t > \beta$ then Meta-Prediction will consider the ELPH prediction reliable, otherwise it replaces the prediction with the opponent's last action. If β is set to 1 then the model will behave similarly to the original Godfather; as β gets closer to zero, the ELPH predictions are used more regardless of their accuracy. How to choose an appropriate value for β is discussed later in the experimental work.

Algorithm 2 is a high-level description of how FAL works.

Algorithm 2 FAL Algorithm

```
Teach_min: is the teaching period
β: is the threshold of confidence in the prediction
while i ≤ end of game do
   if W_t ≤ β then
      Predicted Next Action ← Last Action
   else
      Predicted Next Action ← ELPH Predicted Action
   end if
   if i ≤ Teach_min then
      Play half of Targetable Pair
   else
      Next Action ← Godfather-Future Predicted Next Action
      Compute Average Prediction Success
   end if
   Increment i
end while
```

4. ENSEMBLE OF FAST ADAPTIVE LEARNERS (FAL-ENSEMBLE)

The FAL-Ensemble algorithm (shown as Algorithm 3) functions in a way similar to the original FAL algorithm with a significant difference. It monitors the number of new hypotheses generated in its ELPH predictive model. If at time t the number of new hypotheses reaches a threshold of δ it creates a new predictive model that starts the observations from time t going forward. This process is repeated, every time the number of new hypotheses is above the threshold, a new model is added. When making a prediction FAL-Ensemble takes a majority vote among the ensemble of predictive models weighted by their past accuracy.

Algorithm 3 FAL-Ensemble Algorithm

δ: is the threshold for starting a new model.
$M_1, \ldots M_j$: is the set of ELPH models.
$M_k = \{\text{ELPH}(0)\}$
while $i \leq$ end of game **do**
  if $i \leq Teach_{min}$ **then**
    Next Action $\leftarrow$ Play half of Targetable Pair.
  else
    Next Action $\leftarrow$ Weighted Majority Vote from M_k
  end if
  Added Hypotheses $\leftarrow$ Sum of added hypotheses in M_k
  if Added Hypotheses $\geq \delta$ **then**
    $M_k \leftarrow M_k \bigcup \text{ELPH}(i)$
  end if
  Add New Observation to all models in M_k
  Increment i

end while

5. EXPERIMENTAL RESULTS

We compared experimentally the performance of different learning algorithms, using two-player repeated normal form games as the setting for our experiments. An example of a game matrix is in Table 1, where the row player's payoff is given first followed by the column player's payoff. At each interaction the players play a simultaneous move. After each interaction, each player receives a payoff which depends on the joint action chosen for the interaction.

Each of the normal form games we have chosen was played for 100 iterations. We repeated each of the 100 iterations 100 times to compute average outcomes and reduce the noise. We choose to limit the number of repeated games to 100 to support the core idea of having limited repeated interactions. We also tried the different algorithms against an adversary that at some point switches strategy. This enabled us to explore how the agents respond to situations where the opponent does not converge to a stationary policy.

We have chosen two set of algorithms for our experimental work. Bully and Godfather are representatives of deterministic methods. Q-Learning, WOLF-PHC, and AWESOME represent learning methods. Each one of the learning methods also represents one of the approaches of multiagent learning discussed earlier in section 2 and in [6]. All methods assume knowledge of the underlying game and observe the opponent's actions.

Q-Learning is a classical learning method, WOLF-PHC is one the early algorithms that combine policy search and opponent modeling, AWESOME learns best response. The specific algorithms and strategies we use are:

- Q-Learning is a general purpose semi-supervised learning algorithm that we adapted for the repeated game setting. Q-Learning traditionally assumes a stationary, fully observable environment. In this framework we break these assumption [13].
- WOLF-PHC operates on a different assumption from Q-Learning with the specific notion of deciding the best action depending on whether the agent is losing and winning. WOLF-PHC updates the belief on the current actions with a steady rate as long as it leads to wins. When the action leads to losses, WOLF-PHC changes that belief on the actions with a higher rate in order to find another suitable action. WOLF-PHC converges to the optimal policy under the assumption that the opponent is a learner that will converge to a stationary policy.
- FAL and FAL-Ensemble are our proposed algorithms.
- Godfather is a general class of Tit-For-Tat strategies with the basic idea of giving the opponent the chance to "cooperate" by playing a targetable pair that gains the opponent more than the security value. However if the opponent fails to play the targetable pair, Godfather will punish it by playing the minimax strategy.
- Bully is a deterministic policy that chooses the action the maximizes the player's payoff assuming the opponent will best respond. A more detailed description of Bully is in [13] .
- AWESOME is an adaptive algorithm that learns to play against opponents that (eventually) play a stationary strategy. It does this by observing its opponent's actions and either plays a precomputed Nash equilibrium strategy or a best response strategy.

The experimental work is divided into two groups of experiments. In the first group we test different agents, each one using one of the algorithms in the set listed above. The agents play in pairwise runs against each other in each game. The second group involves a subset of agents this time playing against a specific agent that switches strategy. The goal of this second group of experiments is to understand how fast different types of agents adapt to an opponent that switches its strategy during the game. For FAL, the β is set at 0.75 unless otherwise stated.

Experiment 1

The tables of results presented in this experiment represent the average reward at the end of 100 iterations for each game. Each entry in the table contains two values. The first represents the reward obtained by the row player and the second represents the reward obtained by the column player when playing against each other. We only show the results of FAL because FAL-Ensemble and FAL were identical due to the fact that in FAL-Ensemble the added hypotheses didn't reach the threshold to create an Ensemble.

Chicken

In the game of Chicken, shown in Table 1, in the most cooperative solution each player can get a payoff of 3.0. However, at each interaction the opponent has an incentive to exploit to receive a reward of 3.5. If both agents deviate they both get a lower outcome of 1.0.

3.0,3.0	1.5,3.5
3.5,1.5	1.0,1.0

Table 1: Chicken game matrix.

	Q1	WF	FAL	GF	Bully	AW
Q1	2.4,2.4	2.4,2.4	1.8,2.6	2.5,2.5	1.3,2.5	2.6,2.5
WF		2.4,2.4	1.8,2.6	2.3,2.3	1.3,2.5	2.7,2.4
FAL			3.0,3.0	3.0,3.0	1.0,1.0	2.9,2.9
GF				3.0,3.0	1.0,1.0	3.5,1.5
Bully					1.0,1.0	3.5,1.5
AW						2.5,2.5

Table 2: Average pairwise payoffs after 100 repeated games of Chicken.

Analysis. The reward obtained by Bully is 1.0, which is the lowest reward possible. This happens because Bully assumes the opponent will best respond and so it will never try to cooperate. It is evident that in this game cooperation would yield a higher payoff of 3.0 for both players. Agents that are willing to cooperate, like FAL and Godfather, or are willing to understand when there is potential of higher reward, like Q-learning and WOLF-PHC, end up choosing a set of actions that lead to higher reward, as shown in Table 2. However, an agent like AWESOME doesn't attempt to cooperate and plays the Nash equilibrium strategy to start and it continues to use it against Bully, Godfather, and in self-player. Against WOLF-PHC and Q-Learning, AWESOME switches to a best response strategy when these agents are exploring or cooperating (in the case of FAL). Due to this, AWESOME receives a lower reward then when playing against other agents.

It is important to notice the speed at which the agents were able to get the higher cooperative outcome of 3.0. Let's look at the row of FAL. FAL learns and signals to the opponent that it is willing to cooperate. Q-Learning, while better than Bully, is slow at converging to cooperation and gets only a payoff of 1.8 after 100 games.

Unfortunately, WOLF-PHC and Q-Learning are learners but are relatively slow in learning the cooperative nature of the opponent. The slowness in their learning is evident when compared to the speed of FAL in self-play or even the original Godfather, which is a stationary strategy. Playing against Bully is the most clear example of how the lack of fast learning implies higher reward for the opponent. Bully is a deterministic policy which does not cooperate, but Q-Learning and WOLF-PHC are unable to realize this fast enough and continue to explore in the hope of finding a better outcome. FAL with its fast prediction of the opponent's actions is able to realize this and to revert back to the security value.

Prisoner's Dilemma

In Prisoner's Dilemma, shown in Table 3, the dominant strategy is to defect and receive a reward of 1.0. Cooperating would lead to a higher outcome of 3.0 but with the added risk of getting 0 if the opponent decided to betray.

Analysis. The issue of speed of learning remains a challenge in this game but there are some interesting differences between the nature of the games Chicken and Prisoner's Dilemma that lead to WOLF-PHC and Q-Learning to learn

3.0,3.0	0.0,5.0
5.0,0.0	1.0,1.0

Table 3: Prisoner's Dilemma game matrix.

	Q1	WF	FAL	GF	Bully	AW
Q1	1.7,1.7	1.7,1.7	2.2,2.2	2.4,2.4	0.9,1.4	1.8,1.1
WF		1.9,1.9	2.2,2.2	2.4,2.4	0.9,1.4	2.0,1.1
FAL			3.0,3.0	3.0,3.0	1.0,1.0	1.9,2.0
GF				3.0,3.0	1.0,1.0	1.0,1.0
Bully					1.0,1.0	1.0,1.0
AW						1.0,1.0

Table 4: Average pairwise payoffs after 100 repeated games of Prisoner's Dilemma.

β	WOLF-PHC	FAL
1	2.4	2.4
0.75	2.2	2.2
0.55	2.1	2.1
0.35	2.0	2.0

Table 5: The effect of different β choices for FAL on WOLF-PHC vs FAL in Prisoner's Dilemma.

relatively faster than Bully. In Chicken the difference was between getting a reward of 1.0 versus 1.5 while in Prisoner Dilemma the difference is either 1 or 0 against Bully.

In a side sub-experiment reported in Table 5 we investigated the effect of using different values of β on the performance of our algorithm. This sub-experiment was conducted in Prisoner's Dilemma because it was the game with most significant changes in rewards.

We assume that the benefit occurs because FAL, by trusting its prediction, gives a longer grace period before it punishes the opponent which gives more time to the opponent to understand that FAL is willing to cooperate. However, this result depends on the exploration policy. The grace period might not be sufficient for WOLF-PHC to reach the cooperation stage.

Q-Learning against a simple strategy such as Bully fails to learn that Bully is stationary quickly enough and continues its attempt to explore. This leads to more than 10% loss in reward than the security value. This problem is not apparent in others agents that are adapting faster.

Note that it is evident that Q-Learning in its basic setting has no concept of winning and losing and it is simply learning parameters based on the collected reward. On the other hand, WOLF-PHC has a clear concept of winning and losing and that affects its parameter learning. WOLF-PHC is able to learn at a faster rate than Q-Learning and obtains an average reward of 1.9 in self-play while Q-Learning gets 1.7 in self-play.

AWESOME is able to receive the Nash equilibrium reward against Bully, Godfather, and in self-play and is able to receive a higher reward when playing against FAL, WOLF-PHC, and Q-Learning. It is able to receive a higher reward against these agents because it can take advantage of the initial exploration period of Q-Learning and WOLF-PHC to earn the highest possible reward of 5.0 until Q-Learning and WOLF-PHC eventually converge. Against FAL it is able to receive an even higher reward because FAL initially

3.0,3.0	2.0,0.0
0.0,2.0	1.0,1.0

Table 6: Deadlock game matrix.

	Q1	WF	FAL	GF	Bully	AW
Q1	2.7,2.7	2.7,2.7	2.8,2.8	3.0,3.0	3.0,3.0	2.1,2.1
WF		2.7,2.7	2.8,2.8	3.0,3.0	3.0,3.0	1.8,1.9
FAL			3.0,3.0	3.0,3.0	3.0,3.0	3.0,3.0
GF				3.0,3.0	3.0,3.0	3.0,3.0
Bully					3.0,3.0	3.0,3.0
AW						3.0,3.0

Table 7: Average pairwise payoffs after 100 repeated games of Deadlock.

cooperates and AWESOME is able to earn a reward of 5.0 many times until FAL learns it is doing this and switches to play the Nash equilibrium strategy.

Deadlock

In the game Deadlock, the choice of action is obvious, which is to choose action 1, i.e. cooperating. This leads to a reward of 3.0 for both sides.

Analysis. Deadlock is a simple game where the choices are clear. Most agents choose the cooperative action. The exception is Q-Learning and WOLF-PHC which get rewards slightly below 3.0 due to their exploration, especially when playing against other learning agents. The question remains what is the value of exploration in this game. This is a game where any exploration could be viewed as an irrational choice, because there is no incentive to get a higher reward from the one given. Godfather and Bully are deterministic strategies so they do not change their choice. FAL and AWESOME converge to a cooperative state and do not explore a suboptimal solution while in self-play. WOLF-PHC and Q-Learning continue to explore in self-play and against other learners.

Battle of the Sexes

In the game of Battle of the Sexes both players want to coordinate but they have conflicting interests. There are two pure Nash equilibria where both players choose the same action and one mixed Nash equilibria where both players play their preferred action more often than the opponents preferred action.

3.0,2.0	0.0,0.0
0.0,0.0	2.0,3.0

Table 8: Battle of the Sexes game matrix.

	Q1	WF	FAL	GF	Bully	AW
Q1	2.7,2.1	2.6,2.2	2.9,1.9	3.0,2.0	3.0,2.0	1.5,1.1
WF		2.7,2.0	2.9,2.0	2.9,2.1	2.9,2.0	1.5,1.2
FAL			2.0,3.0	2.0,3.0	2.0,3.0	2.0,3.0
GF				2.0,3.0	2.0,3.0	2.0,3.0
Bully					2.0,3.0	2.0,3.0
AW						2.0,3.0

Table 9: Average pairwise payoffs after 100 repeated games of Battle of the Sexes.

Analysis. In Battle of the Sexes, Godfather and Bully simply play one of the cooperative actions. FAL starts by playing one of the cooperative actions and learns what the opponent strategy is. However, it does not switch strategies because it will not gain any reward by doing so. In fact, if FAL were to switch strategies it would force both itself and its opponent to lose out on all potential reward. WOLF-PHC and Q-Learning receive slightly below the equilibrium reward due to the exploration period. AWESOME best responds to WOLF-PHC and Q-Learning during this time and is able to obtain more reward than it would if it were to not switch from the Nash equilibrium strategy. However, the average reward AWESOME receives is low because it is slow to recognize that its opponent has switched strategies and AWESOME receives a reward of 0.0 for a few iterations.

Collaboration

In the Collaboration game both players want to coordinate and choose the same action and don't have a preference for which action they both agree on. There are two pure Nash equilibria where both players choose the same action.

3.0,3.0	2.0,2.0
0.0,0.0	3.0,3.0

Table 10: Collaboration game matrix.

	Q1	WF	FAL	GF	Bully	AW
Q1	2.8,2.8	2.7,2.7	2.9,2.9	2.9,2.9	2.9,2.9	2.5,2.5
WF		2.7,2.7	2.9,2.9	2.9,2.9	3.0,3.0	2.4,2.4
FAL			3.0,3.0	3.0,3.0	3.0,3.0	3.0,3.0
GF				3.0,3.0	3.0,3.0	3.0,3.0
Bully					3.0,3.0	3.0,3.0
AW						3.0,3.0

Table 11: Average pairwise payoffs after 100 repeated games of Collaboration.

Analysis. In Collaboration, Godfather, Bully, and FAL play one of the cooperative actions and FAL learns that it should not change actions because it will receive less reward even though it knows what action its opponent will play. WOLF-PHC and Q-Learning again receive slightly below the equilibrium reward, similar to Battle of the Sexes, due to the exploration period. AWESOME best responds but the average reward is higher for this game because AWESOME does not receive a reward of 0.0 as often since it now also may receive a reward of 2.0.

Coordination

In the Coordination game both players want to coordinate and choose the same action and they both prefer action 1 over action 2. There are two pure Nash equilibria where both players choose the same action however action 1 dominates the other.

Analysis. In Coordination, the results are similar to Collaboration however the cases when Q-Learning plays against AWESOME and when WOLF-PHC plays against AWESOME are interesting. Both players receive far lower than the equilibrium reward. This is because Q-Learning and WOLF spend time exploring and as such sometimes play action 2. This will initially give both players a reward of 1.0. AWESOME eventually determines that its opponent is

3.0,3.0	0.0,0.0
1.0,1.0	2.0,2.0

Table 12: Coordination game matrix.

	Q1	WF	FAL	GF	Bully	AW
Q1	2.3,2.3	2.2,2.2	2.4,2.4	2.7,2.7	2.3,2.3	1.3,1.3
WF		2.2,2.2	2.6,2.6	2.7,2.7	2.4,2.4	1.3,1.3
FAL			3.0,3.0	3.0,3.0	3.0,3.0	3.0,3.0
GF				3.0,3.0	3.0,3.0	3.0,3.0
Bully					3.0,3.0	3.0,3.0
AW						3.0,3.0

Table 13: Average pairwise payoffs after 100 repeated games of Coordination.

playing a stationary strategy and plays its best response to this strategy which is to play action 2 instead of its initial Nash equilibrium action 1. This forces both players to receive a reward of 2 and the joint strategy of both players playing action 2 is a Nash equilibrium so both players never deviate from this strategy from then on.

5.1 Experiment 2: Switching Strategy

In Experiment 1 we have shown that FAL is able to learn faster, adapt and achieve better results than Q-Learning, WOLF-PHC, and Bully. FAL was also able to receive better results in self-play in comparison to AWESOME in self-play. However, the performance of Godfather, and FAL are almost identical in many scenarios. Moreover, we would like to further show the power of FAL-Ensemble over the original FAL. In order to show the importance of fast adaptive learning we present what happens against an opponent that changes its strategy after some period of time. Detecting the change and adapting to it is the real advantage that we are aiming at achieving in this work.

We introduce a new agent we call *Switch agent*. The agent starts by following the classical Godfather strategy until it reaches stage 40 of the game. After that, the agent follows a deterministic repeated sequence of actions $\{a_1, a_2, a_1, a_1, a_2, a_1\}$ indefinitely. This agent is intended to be deterministic and predictable with a bounded memory. The choice of making the agent switch to a deterministic policy was made to simplify the analysis. Despite its simplicity, learning to play against this opponent is challenging.

In this experiment, we show the results of the Switch agent playing against Godfather, WOLF-PHC, FAL, and FAL-Ensemble in the game of Chicken. The rest of the games show similar trends as Chicken so the analysis would be similar. WOLF-PHC and Godfather were chosen to represent the learner agents because of their strong performance in Experiment 1. Figure 1 shows the average reward over time for the 4 agents against the Switch agent.

Analysis of Rewards Figure 1 shows the graph for the Switch agent against the four agents Godfather, WOLF-PHC, FAL, and FAL-Ensemble. The figure reports the difference Delta in average reward between the agents and the Switch agent. Positive Delta rewards imply Switch is getting more reward and 0 is a tie. FAL and Godfather were able to detect that the opponent is cooperative and converged to a stable reward. WOLF-PHC attempted to learn the cooperative nature but started by fluctuating and did not actually converge to constant reward as FAL and Godfather.

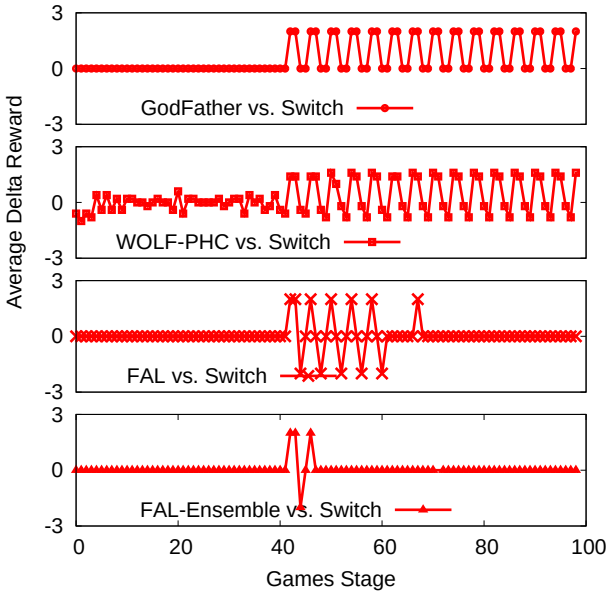

Figure 1: Average delta reward for the 4 Agents vs. Switch agent. Positive values imply the Switch agent is getting more reward, 0 are ties, and the positive values are the others.

At stage 40 of the game, the Switch agent switches to its deterministic strategy. At that stage, the behaviors of the four agents vary. FAL goes on a period of attempting to predict the opponent's action. In this period FAL's prediction accuracy fluctuates which leads into a phase where FAL will use a combination of its prediction or the opponent's previous action to decide its own next action. This continues until FAL succeeds in predicting the Switch agent action sequence which will occur at around stage 60. From that point forward, FAL has a 100% accurate prediction of Switch that will lead to stable Delta rewards. FAL-Ensemble starts a new Predictive model at time 40 and is able to capture the new behaviors without any biases in less than 8 interactions in comparison to 20 in FAL.

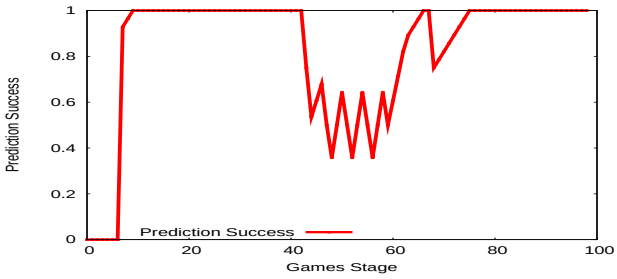

Figure 2: FAL success at predicting the agent's next action.

Figure 2 shows the success of FAL in predicting the action of the Switch agent. The graph shows how FAL after the first 7 interactions is able to determine the next action of its opponent with 100% accuracy. When it reaches stage 40, the prediction success dropped and continued to fluctuate around 40% and 70% but in around 20 interactions it

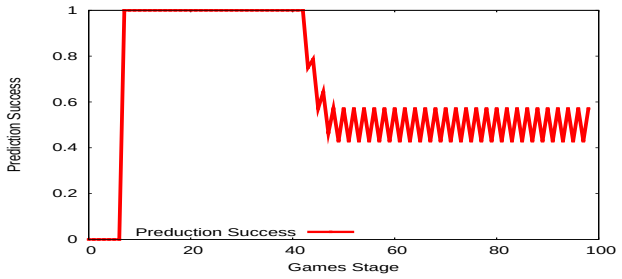

Figure 3: Godfather success at predicting the agent's next action (Godfather prediction is simply the opponent's last action).

went up to almost 100%. This is very significant as it shows that our predictive model is capable of adapting rapidly to changes in the opponents behavior.

Stage 40 and beyond shows a shortcoming of the Godfather strategy. Godfather continues to miscalculate the next action and ends up either getting 1.5 or 3.0, which is driven by the actions of the Switch agent. This issue rises because Godfather makes its next action based on the last action of the opponent while in order to fully capture the opponent a model needs to look at least three past actions as a sequence. Figure 3 shows by using the opponents last action as prediction for its next action Godfather was able to predict the first 40 stages. However, after stage 40 Godfather's prediction is almost random. This also explains our requirement of a predictive model that is able to find and exploit any information hidden in the ordered sequence of actions.

WOLF-PHC struggles in a fashion similar to Godfather after stage 40, by alternating between rewards of 1.5 and above 3.0. The issue with WOLF-PHC that even if WOLF-PHC is able to quickly learn the probability of each action, it will not be a clear sequence predicted as provided by FAL. This will cause WOLF-PHC to have sub-optimal best response to the Switch agent in the limited interactions as shown in the delta rewards. The comparison of these results is important as it shows the true power of FAL in comparison to both WOLF-PHC and Godfather. Moreover, it shows the power of FAL- Ensemble by eliminating bias and add a new predictive model to the Ensemble.

6. CONCLUSIONS AND FUTURE WORK

Our goal in this work was to motivate and introduce the need for new requirements on multiagent learning algorithms. We want to be able to build agents that learn even when interacting with an opponent for a limited number of repeated interactions as well as that have the capability to adapt to sudden and frequent changes in the opponent's strategy regardless of whether the opponent is truly adaptive or even stationary in the limit. We proposed a new algorithm, FAL, and its variation, FAL-Ensemble, which creates a new predictor whenever the opponent strategy switches and which makes predictions by a majority vote among its predictive models. We showed experimentally that FAL and FAL-Ensemble outperform other algorithms in this context. Future work will be directed at examining theoretical properties of FAL, extending it to work with n-player games, making predictions for more than the next opponent's action, and applying it to a larger class of games.

7. REFERENCES

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

[2] B. Banerjee and J. Peng. Performance bounded reinforcement learning in strategic interactions. In *Proc. AAAI Conference*, pages 2–7, 2004.

[3] B. Banerjee and J. Peng. Efficient no-regret multiagent learning. In *Proc. AAAI Conference*, volume 20, pages 41–46, 2005.

[4] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[5] R. Brafman and M. Tennenholtz. Optimal efficient learning equilibrium: Imperfect monitoring in symmetric games. In *Proc. AAAI Conference*, pages 726–731. AAAI Press, 2005.

[6] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, 38(2):156–172, 2008.

[7] D. Chakraborty and P. Stone. Online multiagent learning against memory bounded adversaries. In *Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 211–226. Springer, 2008.

[8] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1):23–43, 2007.

[9] M. Elidrisi and M. Gini. When speed matters in learning against adversarial opponents (Extended Abstract). In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, 2012.

[10] S. Jensen, D. Boley, M. Gini, and P. Schrater. Non-stationary policy learning in 2-player zero sum games. In *Proc. AAAI Conference*, pages 789–794. AAAI Press, 2005.

[11] M. Johanson and M. Bowling. Data biased robust counter strategies. In *Twelfth Int'l Conf. on Artificial Intelligence and Statistics*, pages 264–271, 2009.

[12] M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. Int'l Conf. on Machine Learning*, 1994.

[13] M. Littman and P. Stone. Leading best-response strategies in repeated games. In *Int'l Joint Conf. on Artificial Intelligence Workshop on Economic Agents, Models, and Mechanisms*, 2001.

[14] R. Powers and Y. Shoham. Learning against opponents with bounded memory. In *Proc. Int'l Conf. on Artificial Intelligence*, pages 817–822, 2005.

[15] R. Powers, Y. Shoham, and T. Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1):45–76, 2007.

[16] Z. Wang, A. Boularias, K. Mülling, and J. Peters. Balancing safety and exploitability in opponent modeling. In *Proc. AAAI Conference*, pages 1515–1520, 2011.