Convergence of Strategies in Simple Co-Adapting Games

Richard Mealing and Jonathan L. Shapiro
Machine Learning and Optimisation Group
School of Computer Science
University of Manchester
M13 9PL, UK
{mealingr,jls}@cs.man.ac.uk

ABSTRACT

Simultaneously co-adapting agents in an uncooperative setting can result in a non-stationary environment where optimisation or learning is difficult and where the agents' strategies may not converge to solutions. This work looks at simple simultaneous-move games with two or three actions and two or three players. Fictitious play is an old but popular algorithm that can converge to solutions, albeit slowly, in self-play in games like these. It models its opponents assuming that they use stationary strategies and plays a best-response strategy to these models. We propose two new variants of fictitious play that remove this assumption and explicitly assume that the opponents use dynamic strategies. The opponent's strategy is predicted using a sequence prediction method in the first variant and a change detection method in the second variant. Empirical results show that our variants converge faster than fictitious play. However, they do not always converge exactly to correct solutions. For change detection, this is a very small number of cases, but for sequence prediction there are many. The convergence of sequence prediction is improved by combining it with fictitious play. Also, unlike in fictitious play, our variants converge to solutions in the difficult Shapley's and Jordan's games.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms, Experimentation, Performance, Theory

Keywords

Self-play convergence; opponent modelling; sequence prediction; change detection; fictitious play; Nash equilibrium; iterated normal-form games; empirical distribution

FOGA'15, January 17–20, 2015, Aberystwyth, United Kingdom.

http://dx.doi.org/10.1145/2725494.2725503.

1. INTRODUCTION

Evolutionary computation techniques, such as evolutionary algorithms, swarm intelligence methods, artificial immune systems, etc, often focus on co-adapting agents to solve a shared optimisation problem in a static environment. This is like a game where each agent shares the same reward function, which returns higher rewards for better optimisations. However, in many problems each agent has its own optimisation problem or reward function, which can depend on the other agents' strategies. In these uncooperative cases, co-adapting agents can result in a non-stationary environment making optimisation or learning difficult because the optimal strategies are changing. Examples include auction bidding agents, poker-playing agents, competing agents placing advertisements on web pages, and so forth.

A population of simultaneously co-adapting or coevolving agents in an uncooperative setting may converge or exhibit complex dynamics [37, 12, 13, 14, 38, 9, 7, 8, 2, 11, 44, 6, 23]. The goal of this work is to address the question of whether convergence is enhanced if each agent assumes that the other agents are changing their strategies over time. We study this using simultaneous-move games with two or three actions and two or three players. We compare fictitious play, which is an adaptive mechanism that assumes that the opponent uses a stationary strategy, with two new variants that remove this assumption and explicitly assume that the opponent uses a dynamic strategy. The opponent's strategy is predicted using a sequence prediction method in the first variant and a change detection method in the second variant.

Ideally, we want each agent in a multiagent system to consistently learn and change its strategy to increase its expected rewards. If an agent did this, then eventually it would learn and converge to a best-response strategy that maximises its expected rewards against the other agents' strategies. If all agents did this, then eventually they would learn and converge to a Nash equilibrium. This is why much of the literature about learning in multiagent systems searches for learning rules that will result in agents' strategies converging to a Nash equilibrium. At each step in our approach, we observe the opponent's action, predict its strategy, and play a best-response strategy to the predicted strategy.

We specifically compare the convergence in self-play of our variants and fictitious play to mixed strategy Nash equilibria. Only the empirical distributions of plays over games are considered because they almost always play pure strategies. Our convergence results are purely experimental and find that our variants converge faster than fictitious play. However, unlike in fictitious play, our variants do not always

exactly converge to the Nash equilibria. For change detection, this is a very small number of cases, but for sequence prediction there are many. Combining sequence prediction with fictitious play improves its convergence, reducing these cases. Also, unlike in fictitious play, our variants converge to the mixed strategy Nash equilibria in Shapley's and Jordan's games, which are considered difficult [29].

2. CONVERGENCE TO SOLUTIONS

2.1 Solution Concepts

In an n-player finite normal-form game each player, $i \in \{1, 2, \ldots, n\}$, has a finite set of actions (or pure strategies), $A_i = \{a_1, a_2, \ldots, a_{|A_i|}\}$, and a utility function that maps tuples of actions, where each tuple contains one action per player, to rewards, $u_i : \prod_{j=1}^{n} A_j \to \mathbb{R}$. Each player i also has a strategy, $\sigma_i \in \Sigma_i \equiv \Delta(A_i)$, where $\Delta(\cdot)$ is the space of probability distributions over a set. We define the strategy profile, σ, as the tuple containing each player's strategy, $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \in \Sigma = \prod_{j=1}^{n} \Sigma_j$, and σ_{-i} as the same as σ but excluding player i's strategy, $\sigma_{-i} = (\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n) \in \Sigma_{-i} = \prod_{j=1, j\neq i}^{n} \Sigma_j$. Finally, we define player i's expected reward for the strategy profile σ as $\overline{u}_i(\sigma) \equiv \sum_{a \in \prod_{j=1}^{n} A_j} u_i(a) \prod_{k=1}^{n} \sigma_k(a(k))$, where $a(k)$ is player k's action in a. When playing, each player i simultaneously chooses an action according to their strategy σ_i producing the tuple $a \in \prod_{j=1}^{n} A_j$ and gets a reward of $u_i(a)$. Our definitions follow those by Fudenberg and Levine [22].

Typically, we want to learn a best-response strategy to maximise a player's expected rewards.

Definition 1. A best-response strategy for player i, $\sigma_i^* \in \Sigma_i$, would give it its most preferred outcome against all other players' strategies, $\sigma_{-i} \in \prod_{j=1, j\neq i}^{n} \Sigma_j$, such that

$$u_i(\sigma_i^*, \sigma_{-i}) = \max_{\sigma_i \in \Sigma_i} u_i(\sigma_i, \sigma_{-i}) \tag{1}$$

Note that for a mixed best-response strategy, all pure strategies with non-zero probabilities have equal expected rewards to each other and to the mixture. This must be true because otherwise the pure strategy with a lower (higher) expected reward than the mixture could be chosen less (more) often, creating a strategy with a higher expected reward, meaning that the original strategy was not a best-response strategy.

If all agents are playing a best-response strategy, then they are mutually playing a Nash equilibrium. A Nash equilibrium is a solution concept of a non-cooperative game with two or more players that was proposed by Nash in 1950 [33].

Definition 2. A Nash equilibrium is a strategy profile, $\sigma^* \in \Sigma$, where each player's strategy, $\sigma_i^* \in \Sigma_i$, is a best-response strategy to the other players' strategies such that

$$u_i(\sigma_i^*, \sigma_{-i}) \geq u_i(\sigma_i, \sigma_{-i}) \text{ for all } \sigma_i \in \Sigma_i \text{ and } i \in \{1, 2, \ldots, n\} \tag{2}$$

Nash proved that if players can use mixed strategies, then at least one Nash equilibrium exists for all n-player games, where each player has a finite number of pure strategies [33].

Although players may not have high expected rewards at a Nash equilibrium, each player is playing optimally given that the other players do not change their strategies. Crawford showed that it is difficult to converge to a Nash equilibrium despite highly favourable settings e.g. simple games, two-players, noiseless feedback, infinite repeats, a unique Nash equilibrium, etc. He found that if the agents adapted their strategies using gradient ascent on their expected rewards, then they would fail to converge under these settings in zero-sum normal-form games [12], general-sum normal-form games [13], and evolutionary games [14]. Thus, a lot of research in multiagent systems looks at developing learning rules that will lead to agents' strategies converging to a Nash equilibrium. In this paper, we focus on a simple, old, but also popular approach called fictitious play.

2.2 Convergence Concepts

Throughout this paper we will be interested in a very weak form of convergence, which we will call *empirical Nash convergence*. The obvious and desirable form of convergence would be one in which the agent's strategy converges to a best-response strategy against agents with stationary strategies, and to a Nash equilibrium against similar learning agents or at least in self-play. If the Nash equilibrium has mixed strategies, then the agents would converge in a statistical sense; they would play stochastically, but from distributions that are converging to stationary distributions that are the Nash equilibrium. Several gradient-based algorithms have been shown to do this in some situations, such as Dahl's algorithm [15, 11, 36], WoLF [9], experience-weighted attraction [23], etc. In other situations these algorithms fail, which is an interesting topic but not the subject of our work.

Consider the traditional version of fictitious play that we use as well as our variants. Each one of these players always plays a best-response strategy to its model. This is almost always a pure best-response strategy and is only mixed if multiple pure best-response strategies to its model exist. Thus, its strategy typically cannot converge to a mixed strategy, but its empirical distribution of plays (pure strategy choices) over games can. If, in a game, the empirical distributions of players who almost always play pure-strategies do converge to a mixed strategy profile (possibly a Nash equilibrium), then this often results in their joint strategy cycling around that profile. We define an empirical distribution, and its convergence to another distribution, as follows. Given a finite set, $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, and an infinite sequence of elements from $\mathcal{A}$, $S = (\alpha^1, \alpha^2, \ldots)$, $\alpha^j \in \mathcal{A}$,

Definition 3. The empirical distribution of S at time t is

$$P_S^t(\alpha_i \in \mathcal{A}) = \frac{1}{t} \sum_{j=1}^{t} [\![\alpha_i = \alpha^j]\!] \tag{3}$$

$$= \left(1 - \frac{1}{t}\right) P_S^{t-1}(\alpha_i \in \mathcal{A}) + \left(\frac{1}{t}\right) [\![\alpha_i = \alpha^t]\!], \tag{4}$$

where $[\![\cdot]\!]$ is the Iverson bracket such that $[\![\phi]\!] = 1$ if the predicate ϕ is true, otherwise $[\![\phi]\!] = 0$. Given a probability distribution over $\mathcal{A}$, $P(\alpha_i \in \mathcal{A})$,

Definition 4. The empirical distribution of S converges to P if for any $\epsilon > 0$, and for any divergence measure $D(\cdot||\cdot)$ between distributions, there exists a time t_ϵ such that $D(P_S^t||P) < \epsilon$ for all times $t > t_\epsilon$.

Finally, given a Nash equilibrium, we define empirical Nash convergence as each player's empirical distribution of plays converging to their strategy in this Nash equilibrium.

2.3 Measuring Nash Equilibrium Convergence

To measure the convergence of a tuple of strategies to a Nash equilibrium, we need to be able to measure the difference between each player's strategy in that tuple and and its corresponding Nash equilibrium strategy. In a normal-form game, each of these strategies can be represented as a discrete probability distribution. Thus, we want to be able to measure the difference between two discrete probability distributions P and Q. To do this, we use the Jensen-Shannon divergence metric because it is a true metric, meaning it is non-negative, zero if P and Q are equal, symmetric, and satisfies the triangle inequality. It is calculated using the Jensen-Shannon divergence, and is based on the Kullback-Leibler divergence. In particular, for each player, we calculate the Jensen-Shannon divergence metric between its empirical distribution of plays and its Nash equilibrium strategy, and we take the average of these values to give an average Jensen-Shannon divergence metric i.e.

$$\overline{D_{JSM}} = \frac{1}{n}\sum_{i=1}^{n} D_{JSM}(\sigma_i||\sigma_i^*), \tag{5}$$

where n is the number of players, σ_i is player i's empirical distribution of plays, and σ_i^* is player i's Nash equilibrium strategy. Here we are assuming that σ_i and σ_i^* can each be represented by a discrete probability distribution, which is the case for a mixed strategy in a normal-form game.

The *Kullback-Leibler divergence* between P and Q, $D_{KL}(P||Q)$, is defined as

$$D_{KL}(P||Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}, \tag{6}$$

where $D_{KL}(P||Q) \geq 0$. This only holds if $P(i) = 0$ whenever $Q(i) = 0$. Also, if $P(i) = Q(i) = 0$, then it is assumed that $0 \ln 0 = 0$. If Q is a uniform distribution, then $D_{KL}(P||Q) = -H(P)$ (i.e. negative Shannon entropy, see Equation (12)).

The *Jensen-Shannon divergence* between P and Q, $D_{JS}(P||Q)$, is defined as

$$D_{JS}(P||Q) = \frac{D_{KL}(P||M) + D_{KL}(Q||M)}{2}, \quad \text{where } M(i) = \frac{P(i) + Q(i)}{2}, \tag{7}$$

and $0 \leq D_{JS}(P||Q) \leq \ln(2)$ if log to the base e is used, or $0 \leq D_{JS}(P||Q) \leq 1$ if log to the base 2 is used.

The *Jensen-Shannon divergence Metric* between P and Q, $D_{JSM}(P||Q)$, is defined as

$$D_{JSM}(P||Q) = \sqrt{D_{JS}(P||Q)}, \tag{8}$$

where $0 \leq D_{JSM}(P||Q) \leq \ln(2)$ if log to the base e is used, or $0 \leq D_{JSM}(P||Q) \leq 1$ if log to the base 2 is used.

3. FICTITIOUS PLAY

3.1 Description

Fictitious play is an algorithm that was originally proposed by Brown in 1951 to explain Nash equilibrium play [10]. It assumes that its opponent is playing a stationary, possibly mixed, strategy and estimates this strategy using a frequentist approach. It then plays a best-response strategy to its estimate i.e. a best-response strategy to its opponent's empirical distribution of plays. If its opponent's strategy is stationary, then as more games are played its estimate becomes more accurate and in turn its best-response strategy becomes more accurate. In an iterated normal-form game, fictitious play would update its estimate of the opponent's strategy using Equation (4), where P_S^t is its estimate of the opponent's strategy at time t, $P_S^t = \tilde{\sigma}_{\text{opp}}^t \in \Sigma_{\text{opp}}$, $\mathcal{A}$ is the opponent's set of actions, $\mathcal{A} = A_{\text{opp}}$, and S is the sequence of opponent actions observed in the games. The factor $1/t$ in Equation (4) is like a learning rate and variants could change this (e.g. geometric fictitious play replaces $1/t$ with a constant $z \in [0, 1]$). Thus, at each iteration t, fictitious play predicts that the opponent will play $\tilde{\sigma}_{\text{opp}}^t$ and therefore plays a best-response strategy to $\tilde{\sigma}_{\text{opp}}^t$, i.e. a strategy $\sigma_{\text{FP}}^* \in \Sigma_{\text{FP}}$ where $u_{\text{FP}}(\sigma_{\text{FP}}^*) = \max_{\sigma_{\text{FP}} \in \Sigma_{\text{FP}}} u_{\text{FP}}(\sigma_{\text{FP}}, \tilde{\sigma}_{\text{opp}}^t)$.

3.2 Convergence of Fictitious Play

Fudenberg and Levine showed that for fictitious play in self-play, *strict* Nash equilibria are absorbing states [20]. This means that in an iterated game, if a strict Nash equilibrium is played at some point, then it will also be played at all subsequent points. For a strict Nash equilibrium, the inequalities in Equation (2) are strict, and so it is always a pure strategy Nash equilibrium. For a weak Nash equilibrium, the inequalities in Equation (2) are equalities, and so it is either a pure or a mixed strategy Nash equilibrium. Thus, if in self-play fictitious play converges to a pure strategy profile, then it must be a Nash equilibrium, and if its empirical distributions of plays converge to some (mixed) strategy profile, then that strategy profile must also be a Nash equilibrium. Finally, the empirical distributions of plays of two fictitious players have been shown to converge to Nash equilibria in self-play in: two-player, zero-sum games [35], two-player, two-action games [31], games with an interior evolutionary stable strategy [25], potential games [32], and certain classes of supermodular games [30, 28, 24].

However, for fictitious players in self-play, their empirical distributions of plays do not always converge to a Nash equilibrium. This has been shown in Shapley's game, a general-sum version of rock-paper-scissors, and Jordan's game, a three-player version of matching pennies, despite it not being true in rock-paper-scissors and two-player matching pennies [37, 27]. Fudenberg and Kreps also showed with their persistent miscoordination example that even if its empirical distribution of plays converges, its expected rewards may differ from the expected rewards of the strategy after convergence [20]. Finally, if multiple Nash equilibria exist and fictitious play converges to one in self-play, then it may not be the "best" Nash equilibrium. In fact, it may be objectively worse than another Nash equilibrium, where some players would be strictly better off. In summary, fictitious players usually cannot converge to mixed strategy profiles (e.g. Nash equilibria) as they almost always play pure strategies, but their empirical distributions of plays can. If these distributions converge to a Nash equilibrium, and they are independent from one another, then their expected rewards will also converge to those at that Nash equilibrium. This last point captures the convergence notion we consider.

3.3 Fictitious Play Example

In this example, we play two fictitious players, in self-play in an iterated game of matching pennies. The row player wins if it matches the action of the column player; otherwise the column player wins. A (loss) win is worth $(-)1$. This

is shown in Table 1. Let heads be represented by 1, tails be represented by -1, x be the mean of the row player's actions, and y be the mean of the column player's actions. So, for example, $x = 0$ would mean that the row player has played equal numbers of heads and tails. The row player updates y after observing each of the column player's actions, and will play its sign. The column player updates x after observing each of the row player's actions, and will play *minus* its sign. The dynamics in expectation, of x and y, obey

$$x(t) = \left(1 - \frac{1}{t}\right) x(t-1) + \left(\frac{1}{t}\right) \text{sign}(y(t-1)), \quad (9)$$

$$y(t) = \left(1 - \frac{1}{t}\right) y(t-1) - \left(\frac{1}{t}\right) \text{sign}(x(t-1)), \quad (10)$$

where $\text{sign}(z) = 1$ if $z > 0$, 0 if $z = 0$, and -1 if $z < 0$. If a player has played equal numbers of heads and tails, then under fictitious play dynamics, its opponent would play a (usually uniform) random action. Thus, the expectation of its opponent's play in this situation is zero. This is why these are the expected dynamics. The players' actions will never converge, because the signs of x and y will never stop changing. However, the means *do* converge, albeit slowly. To illustrate these expected dynamics, we ran 10000 iterations of recurrence relations (9) and (10). The initial values of x and y were each randomly set to either -1 or 1 with equal probability. The results are shown in figures 1a, 1b, 1c, and 1d. Note that many two-player, two-action, zero-sum, normal-form game with a mixed strategy Nash equilibrium will be similar, with x and y measuring the difference of the strategy distribution from the equilibrium strategy.

We view this as a dynamical system. We can say the following about recurrence relations (9) and (10):

1. The origin (Nash) point ($x = 0, y = 0$) is a fixed point.
2. The system cycles towards the origin, by switching strategies, as seen in figures 1a and 1b.
3. The period of a cycle grows linearly with time (see Corollary 1).
4. The amount the system moves towards the origin per cycle decreases inversely with time (see Corollary 2).
5. As a consequence of 3 and 4, the convergence rate is $\Theta(1/\sqrt{t})$, where t is time (iteration) (see Corollary 3).

Point 1 is obvious, Appendix A shows points 2, 3, 4, and 5 through its corollaries to Theorem 1, which postulates the first cycle period and the origin distance after it.

The fact that fictitious play undergoes empirical Nash convergence here, but very slowly, as shown empirically in figures 1a, 1b, 1c, and 1d, as well as theoretically in Appendix A, is part of the motivation of this work. Convergence is very slow because each agent takes an increasingly long time to respond to the change in its opponent's strategy. If an agent could identify its opponent's strategy switches more quickly, then it might converge faster, perhaps optimally like $1/t$ (proven in Appendix B), as well as in more situations. This is the idea that we set out to investigate.

4. RELATED WORK

4.1 Fictitious Play Extensions

In this paper, we only consider the traditional fictitious play algorithm as described in Section 3.1. However, to help put this work into context, we will briefly describe some extensions to it. These extensions can allow fictitious play to model changing opponent strategies, to use mixed strategies, and can improve its convergence to solution concepts. Two popular extensions, by Fudenberg and Levine, are geometric fictitious play and stochastic fictitious play [22].

Geometric fictitious play can model changing opponent strategies. It works by giving bigger weights to more recent opponent actions when updating opponent action probabilities. In comparison to the traditional update, equivalent to Equation (4), the only change is that the factor $1/t$, where t is the iteration, is replaced by a constant $z \in [0, 1]$. The constant z is a "forgetting factor", with higher values placing less weight on past opponent actions.

Stochastic fictitious play can play mixed strategies and has exploration. It does this by smoothing the best-response function i.e. instead of selecting a strategy with the maximum expected reward, strategies are selected with probabilities proportional to their expected rewards. A common approach is for player i to play strategy σ_i with probability

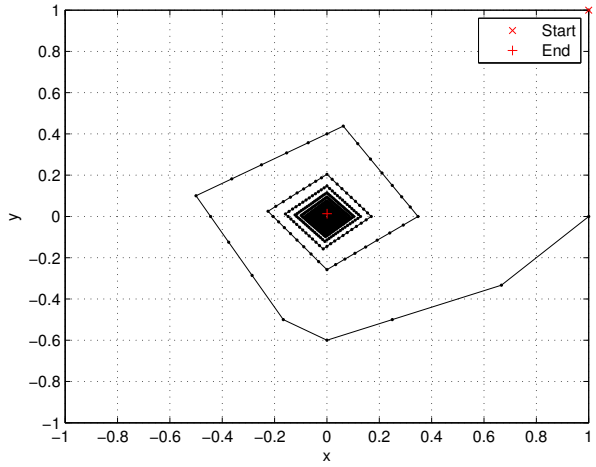
$$\Pr(\sigma_i) = \frac{e^{u_i(\sigma_i, \sigma_{-i})\lambda^{-1}}}{\sum_{\sigma'_i \in \Sigma_i} e^{u_i(\sigma'_i, \sigma_{-i})\lambda^{-1}}}, \quad (11)$$

where λ is a randomisation parameter. As λ approaches zero, this becomes a regular best-response. A similar approach is the κ-exponential fictitious play algorithm [21].

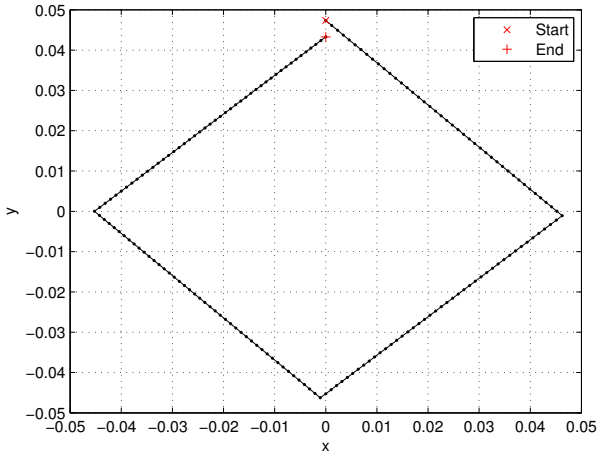
Various extensions are examined by Ny in [34], who looks at traditional (discrete-time) fictitious play, stochastic (smooth) fictitious play, continuous-time fictitious play, and dynamic fictitious play. Two more extensions are proposed by Smyrnakis and Leslie [39, 40], which can model a changing opponent strategy based on recent observations. The first uses a particle filter algorithm, whilst the second uses a heuristic rule to adaptively update the weights of opponent actions.

4.2 Reinforcement Learning

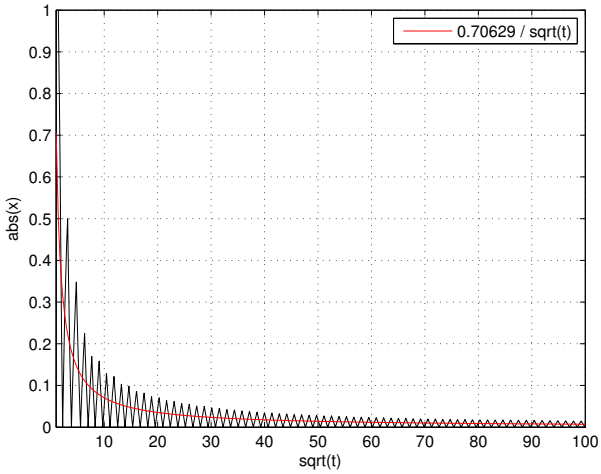
In this paper, we consider convergence to solution concepts, which reinforcement learning can also achieve. An agent using reinforcement learning adapts its strategy based on its rewards. In a game, this implicitly models opponents since rewards are usually determined by them. One group of reinforcement learning methods include gradient ascent, value or policy iteration, and temporal-difference learning. Singh et al. studied agents using gradient ascent on their expected rewards, specifically an Infinitesimal Gradient Ascent (IGA) algorithm, in two-player, two-action, general-sum, iterated normal-form games [38]. They proved that although the agents' strategies may not always converge, their asymptotic average rewards always do converge to the expected rewards of some Nash equilibrium. Dahl proposed the lagging anchor learning model, which draws a player's strategy towards a weighted average of its earlier strategies to improve this convergence [15]. Bowling and Veloso proposed the Win or Learn Fast (WoLF) principle for varying the learning rate, or step-size in the case of gradient ascent, to improve this convergence [9]. The idea is to learn quickly (larger steps) when losing, and slowly (smaller steps) when winning. They proved that WoLF can cause not just the expected rewards, but also the strategies of the gradient ascent agents to converge to those at a Nash equilibrium in two-player, two-action, iterated general-sum games. Several algorithms have been proposed that are based on WoLF including: WoLF-IGA [9], WoLF-PHC (Policy Hill Climb-



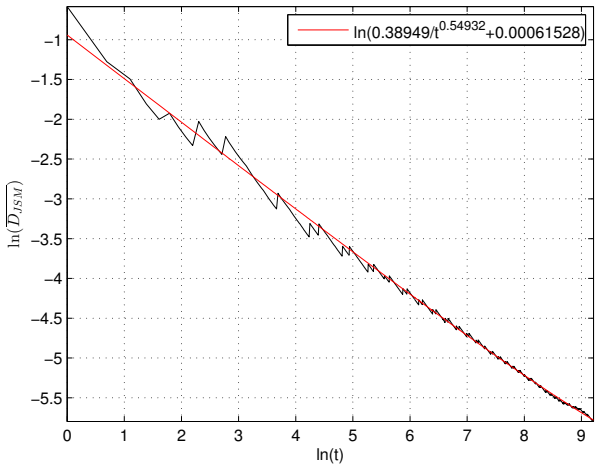
(a) At the start, $x(1) = 1$, $y(1) = 1$, and at the end, $x(10000) = 0.000080$, $y(10000) = 0.014$. They are converging to the Nash equilibrium at the centre, but more slowly as time goes on as the distance between points is decreasing.



(b) One cycle between iterations 866 and 1040. At the start, $x(866) \approx 0$ but positive, $y(866) = 0.047$, and at the end, $x(1040) \approx 0$ but positive, $y(1040) = 0.043$. Note that at iterations 866 and 1040, x has just switched signs i.e. at iterations 865 and 1039 $x \approx 0$ but negative.



(c) The abs(x) is converging towards 0.



(d) The average Jensen-Shannon divergence metric is converging towards 0. See Section 2.3 for a definition of the average Jensen-Shannon divergence metric.

Figure 1: Expected dynamics of fictitious play in self-play in matching pennies over 10000 **iterations. The parameters of the best-fit lines were calculated using MATLAB's Trust-Region-Reflective Least Squares algorithm [1].**

ing) [9], (Policy Dynamics based WoLF) PDWoLF-PHC [7], and (Generalised IGA) GIGA-WoLF [8].

Abdallah and Lesser proposed a slightly different approach to WoLF, to speed up learning when the gradient changes direction, and to slow down learning when the gradient has the same direction [2]. They proposed an algorithm called Weighted Policy Learner (WPL) based on this idea, and found that it converges faster to a Nash equilibrium and is less sensitive to parameter tuning compared to WoLF-PHC, PDWoLF-PHC and GIGA-WoLF. Zhang and Lesser proposed augmenting IGA with policy prediction (IGA-PP) by using gradients for anticipated strategies [44]. They proposed a practical version of this algorithm called Policy Gradient Ascent with Approximate Policy Prediction (PGA-APP), and found that it converges to a Nash equilibrium more quickly and in more situations than WoLF-PHC, WPL, and GIGA-WoLF. A shared feature between WoLF-PHC, GIGA-WoLF, PDWoLF-PHC, WPL, and PGA-APP is that they use Q-Learning [42] to estimate their rewards. More recently, Awheda and Schwartz have proposed using the Exponential Moving Average (EMA) mechanism to update a Q-Learning agent's policy, and have empirically shown that it converges to a Nash equilibrium in more situations than WoLF-PHC, GIGA-WoLF, WPL, and PGA-APP [6].

Adversarial bandits are a second group of reinforcement learning methods that learn using regrets, which are based on rewards. In an adversarial multi-armed bandit problem, at each time step, an adversary sets a reward for each one of k arms, the player selects an arm, and gets its reward. The player's goal is to maximise its rewards over all time steps. This problem is like learning to play in an iterated normal-form game but without prior knowledge of your utility function and against an opponent who knows their utility function as well as your strategy at each iteration. Auer et al. proved that, in this adversarial setting, for T plays the best rate that the per-round reward of an algorithm can approach that of the best-arm is $O(1/\sqrt{T})$ [5]. Many algorithms have been proposed to tackle this problem. A popular example is by Auer et al. who proposed the Exploration-Exploitation with Exponential weights (Exp3) algorithm [5]. It selects actions according to a mixture between a uniform distribution (exploration) and a Gibbs distribution based on the empirical importance-weighted rewards of the arms (exploitation). Another example is by Audibert and Bubeck who proposed the Implicitly Normalised Forecaster (INF) algorithm [4]. It assigns a probability to each action as a function of its estimated regret. Different parameters allow it to be reduced to Exp3, or to give an improved regret upper bound compared to Exp3. The adversarial bandit problem uses pessimistic assumptions and although the algorithms that tackle it have bounded regret, and thus guarantees on their rewards, other algorithms not based on these assumptions may get higher rewards. In particular, if we accurately modelled the opponents and played a best-response strategy to these models at each time step, then we would get higher rewards.

Population-based coevolutionary algorithms are a third group of reinforcement learning methods that learn by aggregating outcomes from interactions between evolving entities. They are stochastic search methods that can find or approximate solutions in interactive domains like games. In biology, coevolution is co-adaptation between distinct populations, but in evolutionary computation it is also co-adaptation within a population. An example is the Nash memory mechanism of Ficici and Pollack [19], which like our sequence prediction method, relies on a memory to learn. It can learn a mixed strategy that monotonically approaches a Nash equilibrium strategy. It works by maintaining two sets of pure strategies. The first set has unbounded size and is the support set for a mixed strategy that is secure against its support set (i.e. its expected payoff is zero or positive against each strategy in its support set). The goal is for the mixed strategy to be secure against strategies that an external search heuristic finds. The second set has finite size and acts as a memory containing strategies that may be useful.

5. OUR FICTITIOUS PLAY VARIANTS

We compare fictitious play with two new variants, which do not assume that the opponent uses a stationary strategy. The opponent's strategy is predicted by the first variant using a sequence prediction method, and by the second variant using a change detection method.

5.1 Sequence Prediction

One approach to opponent modelling is to use a Markov model. A Markov model is a stochastic model that assumes that the Markov property holds. This property holds if the probability of the future depends only on the immediate past i.e. $\Pr(b^{t+1}|b^1, b^2, \ldots, b^t) = \Pr(b^{t+1}|b^t)$. When applied to opponent modelling, the assumption is that the probability of an opponent's action, a^{t+1}_{opp}, only depends on information from the previous iteration. This can be expressed as $\Pr(a^{t+1}_{\text{opp}}|I^1, I^2, \ldots, I^t) = \Pr(a^{t+1}_{\text{opp}}|I^t)$, where I^t is the information available to the agent at time t.

A sequence prediction method uses a model that does not assume that the Markov property holds. Instead, it assumes that the probability of the future can, in general, depend on any subset of the past i.e. $\Pr(b^{t+1}|b^1, b^2, \ldots, b^t) = \Pr(b^{t+1}|H)$ where $H \subseteq \{b^1, b^2, \ldots, b^t\}$. When applied to opponent modelling, the assumption is that the probability of an opponent's action, a^{t+1}_{opp}, can depend, in general, on any subset of information from past iterations. This can be expressed as $\Pr(a^{t+1}_{\text{opp}}|H)$ where $H \subseteq \{I^1, I^2, \ldots, I^t\}$.

Sequence prediction methods usually have two components, a short-term memory, and a long-term memory. The short-term memory, S, is an ordered sequence of the previous $k \in \mathbb{Z}$ observations i.e. $S = (b^{t-k+1}, b^{t-k+2}, \ldots, b^t)$ where b^t is the observation at time t. The long-term memory, L, is a map from sequences of observations and observations to counts i.e. $L : (b_1, b_2, \ldots, b_i) \times B \to \mathbb{Z}$, where b_i is the i-th symbol in the sequence, $0 \leq i \leq k$, and B is the set of values an observation can take. These mappings can be used to form conditional probability distributions such that the probability of an observation, b, given a sequence of observations, S', is the count of that observation given that sequence, $L(S', b)$, divided by the sum of the counts of any observation given that sequence i.e. $\Pr(b|S') = L(S', b)/\sum_{b' \in B} L(S', b')$.

5.1.1 Entropy Learned Pruned Hypothesis space

We use the Entropy Learned Pruned Hypothesis space sequence prediction method proposed by Jensen et al. [26]. It works as shown in Algorithm 1.

Here, the Shannon entropy of P, $H(P)$, is defined as

$$H(P) = -\sum_i P(i) \ln P(i). \tag{12}$$

Algorithm 1 Entropy Learned Pruned Hypothesis Space

Require: Short-term memory size $k \in \mathbb{Z}$, entropy threshold $(0 \leq H_l \leq 1) \in \mathbb{R}$, and a set of possible observations B

Initialise short and long term memories $S \leftarrow ()$, $L \leftarrow \{\}$
function OBSERVE(an observation b)
  Get set of all subsequences of S, $\mathcal{P}(S) \leftarrow \{(), (S(1)), \ldots, (S(|S|)), (S(1), S(2)), \ldots, (S(1), S(|S|)), \ldots, (S(1), S(2), \ldots, S(|S|))\}$
  for all $S' \in \mathcal{P}(S)$ **do**
    if $(S', b) \notin L$ **then**
      Initialise b count for S', $L(S', b) \leftarrow 0$
    end if
    Increment b count for S', $L(S', b) \leftarrow L(S', b) + 1$
  end for
  if $\overline{H(L(S'))} > H_l$ **then** ▷ High entropy
    Remove counts for S', $L \setminus (S', b')$ for all $b' \in B$
  end if
  Add b to end of S, $S \leftarrow (S, b)$
  if $|S| > k$ **then**
    Remove start of S, $S \leftarrow (S(2), \ldots, S(k+1))$
  end if
end function
function PREDICT
  Get set of all subsequences of S, $\mathcal{P}(S)$
  $S'' \leftarrow \arg\min_{S' \in \mathcal{P}(S)} \overline{H_{\text{rel}}}(L(S'))$ ▷ Low entropy
  return $\Pr(b|S'') = \frac{L(S'', b)}{\sum_{b' \in B} L(S'', b')}$ for all $b \in B$
end function

The reliable Shannon entropy of P is calculated by altering the underlying counts that P is assumed to be based on. Given $P(i) = \frac{c(i)}{\sum_i c(i)}$, where $c(i)$ is the count of i, a single count is added for an unknown and new category. The reliable Shannon entropy of P, $H_{\text{rel}}(P)$, is then defined as

$$H_{\text{rel}}(P) = -\frac{1}{\sum_i c(i) + 1} \ln \frac{1}{\sum_i c(i) + 1} - \sum_i \frac{c(i)}{\sum_j c(j) + 1} \ln \frac{c(i)}{\sum_j c(j) + 1}. \quad (13)$$

The (reliable) Shannon entropy of P has a minimum value of 0 and a maximum value of $\ln(m)$, where m is the number of categories in P. Thus, it can be normalised, the normalised (reliable) Shannon entropy, $\overline{H_{[\text{rel}]}}(P)$, is defined as

$$\overline{H_{[\text{rel}]}}(P) = \frac{1}{\ln(m)} H_{[\text{rel}]}(P). \quad (14)$$

5.2 Change Detection

A change detection method observes a sequence of observations and attempts to identify abrupt changes in the parameters of the underlying probability distribution describing those observations. It may consider if a single change has occurred, or if several changes have occurred, and may try to identify when any change(s) occurred. Any change detection method must trade-off between three metrics: false positive rate, false negative rate, and detection delay. When applied to opponent modelling, the assumption is that the underlying probability distributions describing the opponent's strategy are changing abruptly. The change detection method would then infer when the most recent changes have occurred. Observations prior to the times of these inferred changes can then be given lower weights or discarded when predicting the new distributions.

5.2.1 Bayesian Online Changepoint Detection

We use the Bayesian online changepoint detection method proposed by Fearnhead and Liu [17] as well as by Adams and MacKay [3]. This method allows you to specify a model for the distribution and so we model the opponent's strategy as a categorical distribution using a Dirichlet conjugate prior. It works by calculating a posterior distribution over the runlength, where the runlength is the number of steps since the distribution last changed, and then using it to estimate the sample distribution [17, 3]. It assumes that the sample distribution, conditioned on a particular runlength, can be computed. This allows the marginal sample distribution to be calculated by integrating over its posterior distribution conditioned on the current runlength as follows

$$\Pr(x_{t+1}|x_{1:t}) = \sum_{r_t} \Pr(x_{t+1}|r_t, x_{1:t}) \Pr(r_t|x_{1:t}). \quad (15)$$

Here, r_t is the runlength at time t, x_t is the sample at time t, and $x_{i:j}$ are the samples from time i to time j inclusive. To predict the last changepoint optimally, this method considers all possible runlengths and weights them by their probabilities given the samples. The authors show that exact inference on the runlength can be done using a message passing algorithm. The inference procedure is as follows

$$\begin{aligned} \Pr(r_t|x_{1:t}) &= \frac{\Pr(r_t, x_{1:t})}{\Pr(x_{1:t})} = \frac{\sum_{r_{t-1}} \Pr(r_t, r_{t-1}, x_{1:t})}{\Pr(x_{1:t})} \\ &= \frac{\sum_{r_{t-1}} \Pr(r_t, x_t|r_{t-1}, x_{1:t-1}) \Pr(r_{t-1}, x_{1:t-1})}{\Pr(x_{1:t})} \\ &= \frac{\sum_{r_{t-1}} \Pr(r_t|r_{t-1}) \Pr(x_t|r_{t-1}, x_{1:t-1}) \Pr(r_{t-1}, x_{1:t-1})}{\Pr(x_{1:t})}. \end{aligned}$$

Note that the sample distribution, $\Pr(x_t|r_{t-1}, x_{1:t-1})$, is determined by the most recent data. The derivation just applies the laws or rules of probability (conditional probability and joint probability). The last line assumes that the runlength is independent of the previous samples and only depends on the previous runlength i.e. $\Pr(r_t|r_{t-1}, x_{1:t-1}) = \Pr(r_t|r_{t-1})$. This is a message passing algorithm as r_t can only take values based on r_{t-1}. Specifically, either $r_t = 0$ if a change occurs, or $r_t = r_{t-1} + 1$ if a change does not occur.

The probability $\Pr(r_t|r_{t-1})$ is given by a switching rate or "hazard" function $h(t)$ for both values. A simple approach is to assume that the hazard function returns a constant probability for a change $\Pr(r_t = 0|r_{t-1}) = h(0) = \gamma$. The probability of no change would then be one minus this i.e. $\Pr(r_t = r_{t-1} + 1|r_{t-1}) = h(r_{t-1} + 1) = 1 - \gamma$. The hazard function would return zero for all other values of t. Setting its value is a trade-off; high values decrease the detection delay, but increase the number of false positives/negatives. Conversely, low values increase detection delay, but decrease the number of false positives/negatives. Methods have been proposed by Wilson et al. [43] as well as by Turner et al. [41] to learn the hazard function from the data. The former can learn a hazard function that is piecewise constant using a hierarchical generative model, whilst the latter can learn any parametric hazard function via gradient descent.

The space complexity grows linearly with the number of samples because there is a possible runlength for each sample. The time complexity also grows linearly because each possible runlength requires an update. To place an upper limit on the number of possible runlengths, and in turn the memory requirements, a particle filter is used as suggested by Fearnhead and Liu [17], which maintains a finite sample of the runlength distribution. A particle filter is a Monte-Carlo method that estimates a sequential Bayesian model. Each particle represents a point in the distribution with its weight being its approximate probability. If the number of particles grows too large, then resampling takes place where some particles are thrown away and the weights of the remaining particles are updated. The resampling scheme used is called Stratified Optimal Resampling (SOR). Under this scheme the reweighting ensures that the expected values of the new weights are equal to the original weights. It is optimal in that the expected squared difference between the original and the new weights is minimised. The SOR procedure is shown in Section 3.2 of Fearnhead and Liu [17].

6. RESULTS

In the following experiments, we compare the convergence in self-play of fictitious play to our variants of it. Specifically, since each of the algorithms play pure strategies, we look at the convergence of their empirical distributions of plays (i.e. their empirical Nash convergence). For each game, we measure the distances of their empirical distributions of plays from the unique mixed strategy Nash equilibrium. Distances are measured using the Jensen-Shannon divergence metric as defined in Section 2.3. From these distances, we calculate estimates of their empirical Nash convergence speeds.

The first experiment looks again at matching pennies. The second experiment looks at various two-player, two-action, normal-form games derived from generalised matching pennies. The third experiment looks at Shapley's game. Finally, the fourth experiment looks at Jordan's game. In all of the experiments, the sequence prediction method we use is Entropy Learned Pruned Hypothesis Space (ELPH) by Jensen et al. [26] with a short-term memory size of $k = 1$ and an entropy threshold of $H_l = 1$, whilst the change detection method we use is Bayesian online change detection using a categorical model (BayesCPD-C) with a switching rate or hazard function of $h(0) = 1 \times 10^{-4}$ and 100 particles for Stratified Optimal Resampling (SOR).

In the second, third, and fourth experiments we also test a simple hybrid algorithm that combines sequence prediction with fictitious play to try to improve its empirical Nash convergence. It works by playing a best-response strategy to the distribution predicted by sequence prediction if a category in that distribution has a probability greater than some threshold, P_l, where in these experiments $P_l = 0.95$, otherwise it plays a best-response strategy to the distribution predicted by fictitious play.

6.1 Normal-form Games

6.1.1 Matching Pennies

Matching pennies is a two-player, two-action, zero-sum, normal-form game. Each player's actions are heads or tails. Player one wants to match the coin face of player two, and player two wants to mismatch the coin face of player one. There is a unique mixed strategy Nash equilibrium, which is for each player to play each of its actions with equal probability of 1/2. Table 1 shows its rewards.

Table 1: Matching pennies rewards.

	H	T
H	1,-1	-1,1
T	-1,1	1,-1

6.1.2 Generalised Matching Pennies

We create a variety of two-player, two-action, normal-form games derived from generalised matching pennies. Most are general-sum, and some are zero-sum. In these games, a player's strategy has one parameter, which is the probability of it playing its first action. Let these probabilities be p and q for the row and column players respectively. For each of these games, the rewards are set to those shown in Table 2. This creates a game with a single mixed strategy Nash equilibrium at $(p = p*, q = q*)$, where we can choose $p*$ and $q*$. This is proven in Appendix C. If we set $p* = q*$, then the game is zero-sum, otherwise it is general-sum.

Table 2: Rewards for a two-player, two-action, normal-form game with a Nash equilibrium at $(p*, q*)$, where $p*$ and $q*$ are the row player's and the column player's Nash equilibrium probabilities of playing their first actions respectively.

	C_1	C_2
R_1	$\frac{2}{q*}$ - 3,-$\frac{2}{p*}$ + 3	-1,1
R_2	-1,1	1,-1

6.1.3 Shapley's Game

Shapley's game [37] is a two-player, three-action, general-sum, normal-form game. It is the same as rock-paper-scissors, which is a zero-sum game, but negative rewards are replaced with zero rewards, which turns it into a general-sum game. The unique mixed strategy Nash equilibrium is the same as in rock-paper-scissors, i.e. for each player to play each of its actions with equal probability of 1/3. Shapley showed it as an example of where the empirical distributions of plays of two fictitious players' fail to converge to the Nash equilibrium in self-play [37]. Table 3 shows its rewards.

Table 3: Shapley's game rewards.

	R	P	S
R	0,0	0,1	1,0
P	1,0	0,0	0,1
S	0,1	1,0	0,0

6.1.4 Jordan's Game

Jordan's game [27] is a three-player, two-action, general-sum, normal-form game. It extends matching pennies to include a third player. Each player can select heads or tails. Player one wants to match the coin face of player two, player two wants to match the coin face of player three, and player three wants to mismatch the coin face of player one. The unique mixed strategy Nash equilibrium is for each player to play each of its actions with equal probability of 1/2. Table 4 shows its rewards.

Table 4: Jordan's game rewards. Player 1 chooses the outer row, player 2 chooses the column, and player 3 chooses the inner row.

		H	T
H	H	1,1,-1	-1,-1,-1
	T	1,-1,1	-1,1,1
T	H	-1,1,1	1,-1,1
	T	-1,-1,-1	1,1,-1

6.2 Observations

6.2.1 Matching Pennies

The results for sequence prediction and change detection in matching pennies are shown in Figure 2. They show that the empirical Nash convergence of each method is faster compared to fictitious play in Figure 1. Specifically, comparing their average Jensen-Shannon divergence metrics, each method is converging nearly optimally like $1/t$, whereas fictitious play is converging like $1/\sqrt{t}$. Similarly to fictitious play, their agents' empirical distributions of plays cycle around the Nash equilibrium to some degree, with successive cycles getting smaller. However, the cycles of these methods get smaller more quickly. Change detection, like fictitious play, has cycles that get consistently closer to the Nash equilibrium whereas sequence prediction has more irregular cycles.

6.2.2 Generalised Matching Pennies

The results for a variety of two-player, two-action, normal-form games derived from generalised matching pennies are shown in Figure 3. They show that fictitious play has empirical Nash convergence in all of the games, which is expected theoretically. This is not the case for sequence prediction, which does not have empirical Nash convergence in most cases. In fact, the results for sequence prediction in matching pennies seem to be more of an exception rather than the rule. It seems to converge further away from a Nash equilibrium when at that Nash equilibrium at least one player has a strategy with a large magnitude. Conversely change detection has empirical Nash convergence in almost all cases. The handful of cases where it does not converge are where the Nash equilibrium is for one player to be almost indifferent between its actions, and the other player to be almost certain of its actions. The hybrid algorithm, sequence prediction and fictitious play, improves on sequence prediction by having empirical Nash convergence in more cases. The cases where it does not are where at the Nash equilibrium at least one player has a strategy with a large magnitude.

The results also show estimates for the mean empirical convergence rate, $\overline{b}$, and the mean asymptotic convergence distance from the Nash equilibria, $\overline{c}$, for each method. These estimates are calculated by fitting the equation $\overline{D_{JSM}} = a/t^b + c$ to the results of each game and finding the mean b and c parameters. For fictitious play, $\overline{b} = 0.55$, which corresponds to an empirical convergence rate like $1/\sqrt{t}$. Whereas for sequence prediction, change detection, and sequence prediction with fictitious play, $\overline{b} = 0.93$, $\overline{b} = 0.98$, and $\overline{b} = 0.95$ respectively, which corresponds to a nearly optimal empirical convergence rate like $1/t$. Also for both fictitious play and change detection, $\overline{c} = 0.00$, so they empirically converge to the Nash equilibria on average. Whereas for sequence prediction and sequence prediction with fictitious play, $\overline{c} = 0.09$ and $\overline{c} = 0.01$, so they sometimes empirically converge away from the Nash equilibria.

6.2.3 Shapley's Game

The results for Shapley's game are shown in Figure 4. They show that fictitious play does not have empirical Nash convergence. Its average Jensen-Shannon divergence metric decreases slightly but eventually oscillates around a value away from zero with constant amplitude and an ever increasing period. Change detection follows a similar pattern, except its oscillations decrease in amplitude until they eventually fade out, and its value is much closer to zero such that it essentially has empirical Nash convergence. Sequence prediction with or without fictitious play both have empirical Nash convergence at a nearly optimal rate like $1/t$.

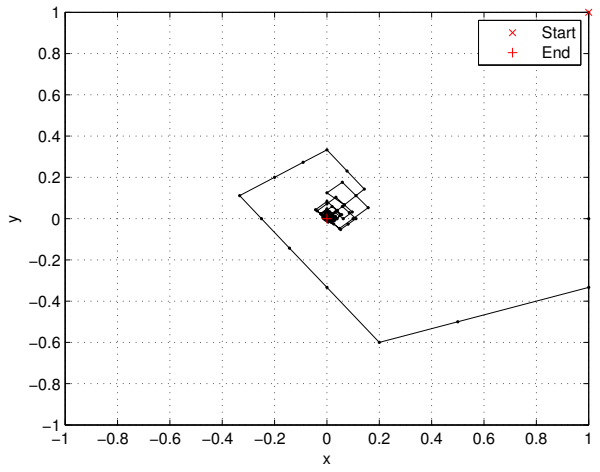
6.2.4 Jordan's Game

The results for Jordan's game are shown in Figure 5. They show that fictitious play does not have empirical Nash convergence. Its average Jensen-Shannon divergence metric oscillates around a value away from zero with constant amplitude and an ever increasing period. But sequence prediction with or without fictitious play as well as change detection have empirical Nash convergence, each at a nearly optimal rate like $1/t$.

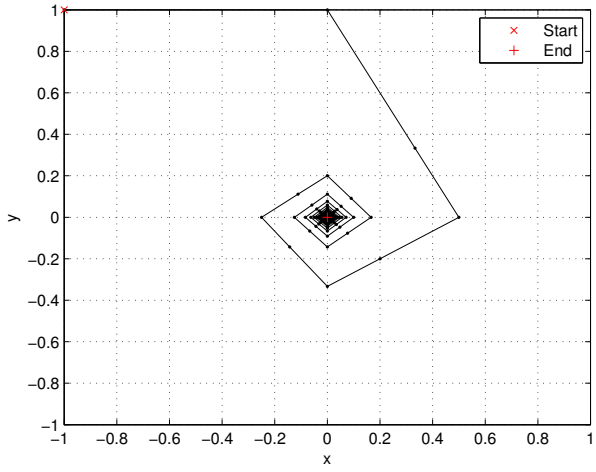
7. CONCLUSIONS

We have proposed two new variants of fictitious play, which assume that the opponents have dynamic strategies. The first variant uses sequence prediction to predict an opponent's strategy based on different contexts of its most recent actions and its empirical distributions of plays that have occurred after these contexts. The second variant uses change detection to infer a distribution over possible changepoints in an opponent's strategy and uses this distribution to predict its strategy. Each variant, like fictitious play, plays a pure best-response strategy to its predicted opponent strategies. We experimentally compared the convergence in self-play of the empirical distributions of plays of fictitious play and our variants to mixed strategy Nash equilibria. The results show that our variants converge faster than fictitious play. However, in generalised matching pennies games, whilst fictitious play and change detection always converge, sequence prediction does not converge in most cases. Also in these games, combining sequence prediction with fictitious play decreases the estimate of its mean convergence distance from Nash equilibria, and increases the estimate of its mean convergence speed. The results also show that, unlike in fictitious play, our variants and the hybrid algorithm converge to the Nash equilibria in Shapley's and Jordan's games, which is known to be difficult. Overall, we find that whilst sequence prediction is somewhat unstable, change detection has better self-play performance than fictitious play in these games.

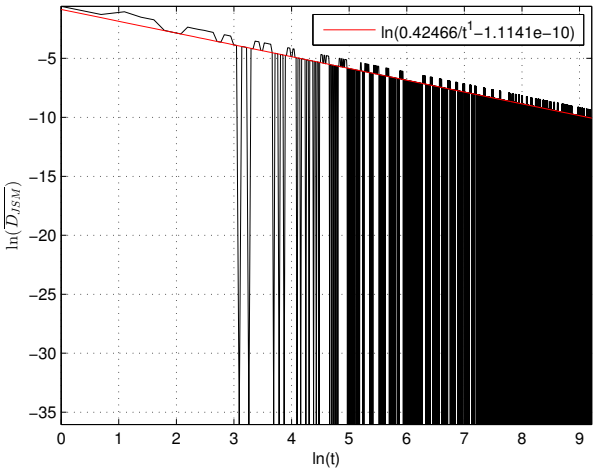
Future work will investigate why our variants converge and how our ideas and results generalise beyond the examined games. We suspect that convergence mainly depends on the amount of history used, if too low, then the agent may have insufficient resolution to predict accurately. For example, using a sliding window of size one, it would always appear as if the opponent will repeat their last action. We also suspect that our ideas and results will generalise to similar normal-form games and situations where fictitious play has been successful like in limit Texas hold'em [16, 18].



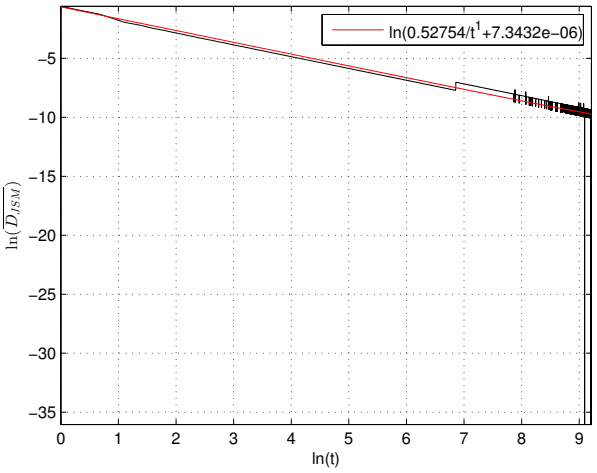
(a) Sequence prediction. At the start, $x = 1$, $y = 1$, and at the end, $x = 0$, $y = 0$. They are converging towards the Nash equilibrium at the centre.



(b) Change detection. At the start, $x = -1$, $y = 1$, and at the end, $x = 0$, $y = 0$. They are converging towards the Nash equilibrium at the centre.

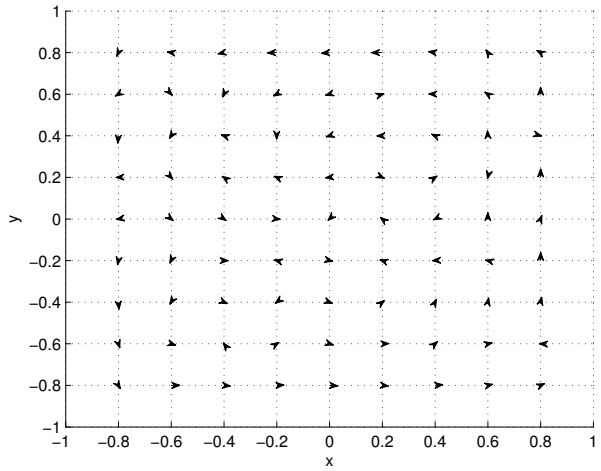


(c) Sequence prediction. The average Jensen-Shannon divergence metric is converging towards 0.

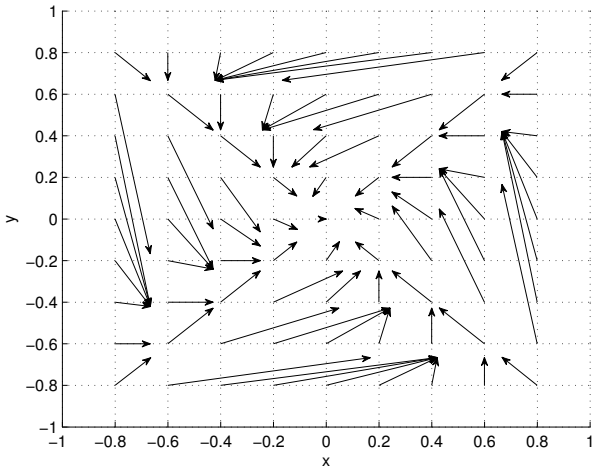


(d) Change detection. The average Jensen-Shannon divergence metric is converging towards 0.

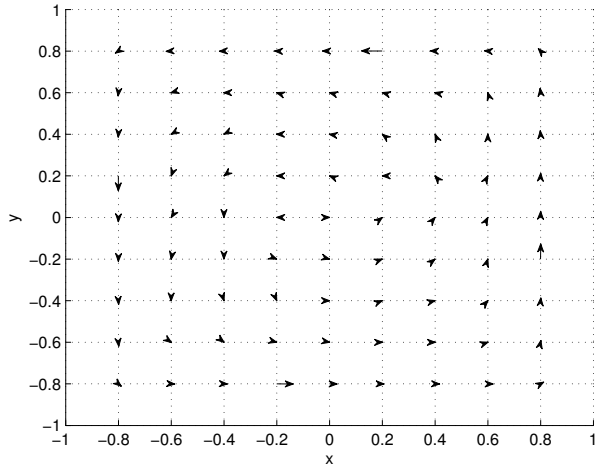
Figure 2: Sequence prediction and change detection each in self-play in matching pennies over 10000 **iterations. The parameters of the best-fit lines were calculated using MATLAB's Trust-Region-Reflective Least Squares algorithm [1].**



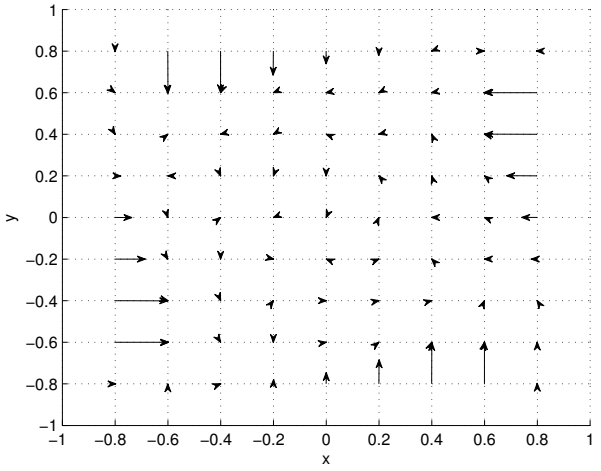
(a) Fictitious play, $\overline{b} = 0.5484$, $\overline{c} = 0.0004$. Each average Jensen-Shannon divergence metric is converging towards 0.



(b) Sequence prediction, $\overline{b} = 0.9307$, $\overline{c} = 0.0923$. Each average Jensen-Shannon divergence metric is converging towards $c \geq 0$, where it tends to be larger if at least one player has a Nash equilibrium strategy with a large magnitude.

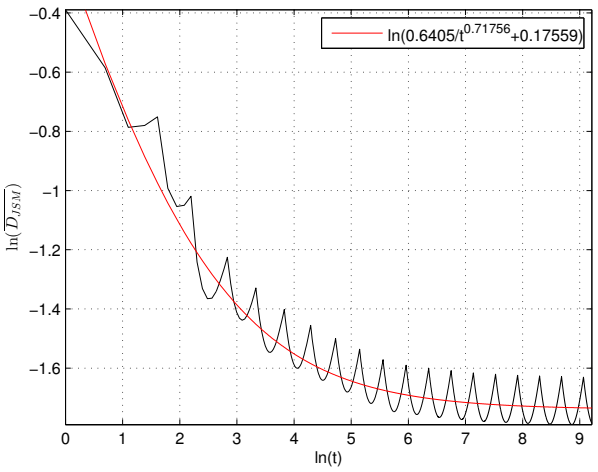


(c) Change detection, $\overline{b} = 0.9829$, $\overline{c} = 0.0010$. Almost all average Jensen-Shannon divergence metrics are converging towards 0. Only a few are converging towards $c > 0$ where one player has a Nash equilibrium strategy near 0 and the other player does not.

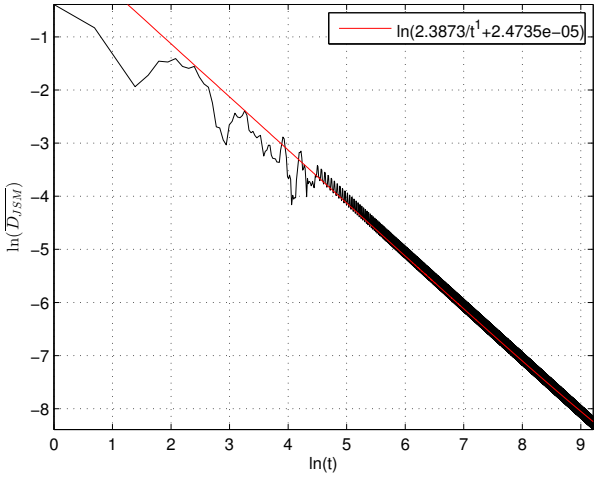


(d) Sequence prediction and fictitious play, $\overline{b} = 0.9546$, $\overline{c} = 0.0088$. Most average Jensen-Shannon divergence metrics are converging towards 0. Some are converging towards $c > 0$ where at least one player has a Nash equilibrium strategy with a large magnitude.

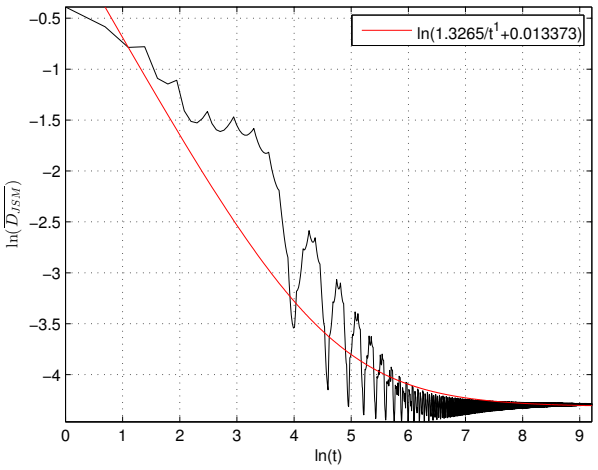
Figure 3: Empirical Nash convergence of various methods in self-play in two-player, two-action, normal-form games with Nash equilibria at positions $\{(x*, y*)|x* \in \{-0.8, -0.6, \ldots, 0.8\}\}, y* \in \{-0.8, -0.6, \ldots, 0.8\}\}$. Each arrow points from a Nash equilibrium position to the position of the method's empirical distribution of plays after 10000 iterations. An estimate for the mean empirical Nash convergence rate, $\overline{b}$, is shown for each method. This is calculated by fitting the equation $\ln(\overline{D_{JSM}}) = \ln(a/t^b + c)$ to the results of each game and taking the average of the b values. The parameters of the best-fit lines were calculated using MATLAB's Trust-Region-Reflective Least Squares algorithm [1].



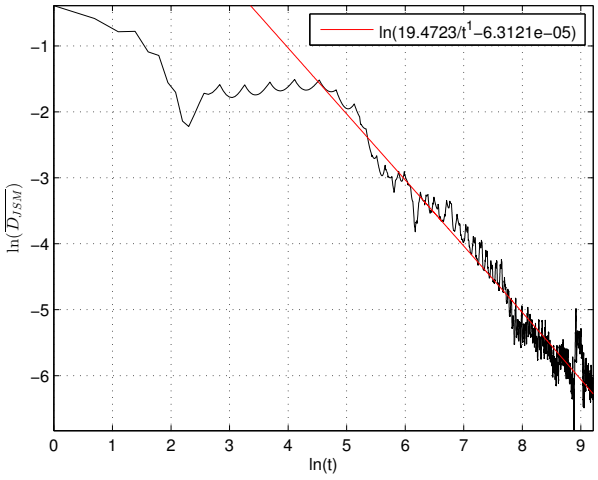
(a) Fictitious play. The average Jensen-Shannon divergence metric is oscillating around a value away from 0.



(b) Sequence prediction. The average Jensen-Shannon divergence metric is converging towards 0.

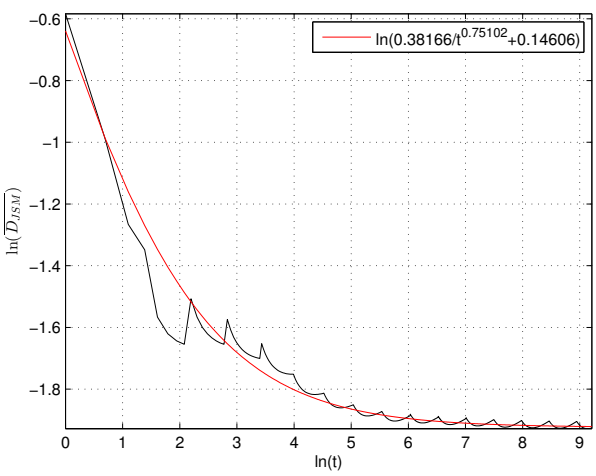


(c) Change detection. The average Jensen-Shannon divergence metric is converging towards a value near 0.

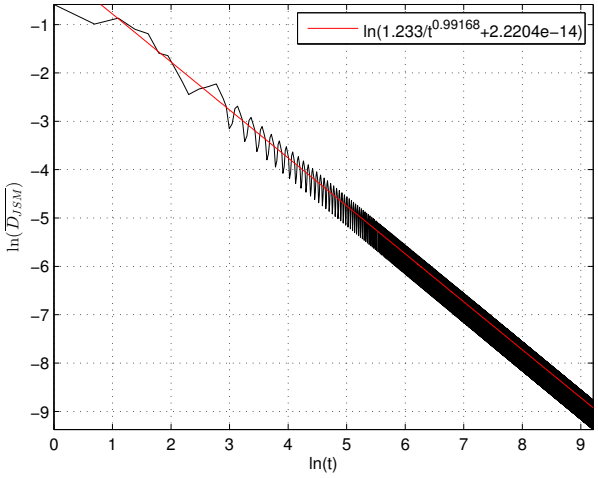


(d) Sequence prediction and fictitious play. The average Jensen-Shannon divergence metric appears to be converging towards 0.

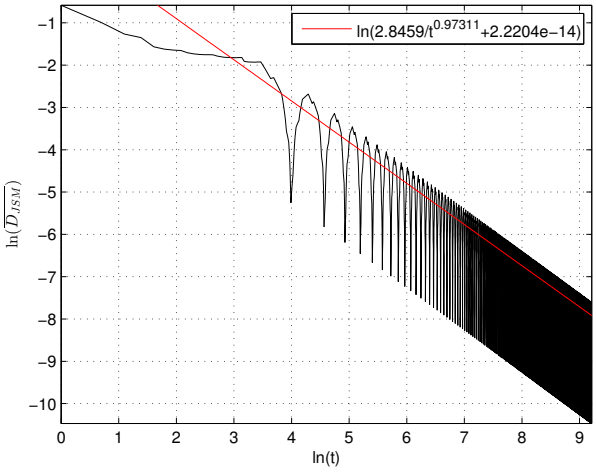
Figure 4: Empirical Nash convergence of various methods in self-play in Shapley's game over 10000 **iterations. The parameters of the best-fit lines were calculated using MATLAB's Trust-Region-Reflective Least Squares algorithm [1].**



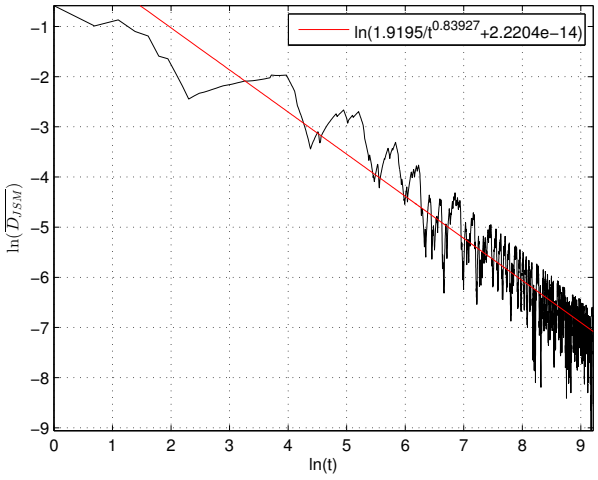
(a) Fictitious play. The average Jensen-Shannon divergence metric is oscillating around a value away from 0.



(b) Sequence prediction. The average Jensen-Shannon divergence metric is converging towards 0.



(c) Change detection. The average Jensen-Shannon divergence metric is converging towards 0.



(d) Sequence prediction and fictitious play. The average Jensen-Shannon divergence metric appears to be converging towards 0.

Figure 5: Empirical Nash convergence of various methods in self-play in Jordan's game over 10000 **iterations. The parameters of the best-fit lines were calculated using MATLAB's Trust-Region-Reflective Least Squares algorithm [1].**

8. ACKNOWLEDGEMENTS

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/P505631/1] and the University of Manchester.

9. REFERENCES

[1] Least-squares algorithms. `http://bit.ly/1rSLrJY`. Accessed: 13/04/2014.

[2] S. Abdallah and V. R. Lesser. Non-linear dynamics in multiagent reinforcement learning algorithms. In *AAMAS 3*, pages 1321–1324, 2008.

[3] R. P. Adams and D. J. C. MacKay. Bayesian online changepoint detection, 2007.

[4] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT 22*, 2009.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multi-armed bandit problem. *SICOMP*, 32:48–77, 2002.

[6] M. Awheda and H. Schwartz. Exponential moving average q-learning algorithm. In *IEEE ADPRL*, 2013.

[7] B. Banerjee and J. Peng. Adaptive policy gradient in multiagent learning. In *AAMAS*, 2003.

[8] M. Bowling. Convergence and no-regret in multiagent learning. In *NIPS 17*, 2005.

[9] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *AI*, 136:215–250, 2002.

[10] G. W. Brown. *Activity Analysis of Production and Allocation*, chapter Iterative Solutions of Games by Fictitious Play, pages 374–376. Wiley, 1951.

[11] J. M. Butterworth and J. L. Shapiro. Stability of learning dynamics in two-agent, imperfect information games. In *10th ACM SIGEVO workshop*. ACM, 2009.

[12] V. Crawford. Learning the optimal strategy in a zero-sum game. *Econometrica*, 42:885–891, 1974.

[13] V. Crawford. Learning behavior and mixed-strategy nash equilibria. *JEBO*, 6:69–78, 1985.

[14] V. Crawford. Learning and mixed-strategy equilibria in evolutionary games. *JTB*, 140:537–550, 1989.

[15] F. A. Dahl. The lagging anchor algorithm: Reinforcement learning in two-player zero-sum games with imperfect information. *ML*, 49:5–37, 2002.

[16] W. Dudziak. Using fictitious play to find pseudo optimal solutions for full-scale poker. In *ICAI*, 2006.

[17] P. Fearnhead and Z. Liu. On-line inference for multiple change points problems. *JRSS B*, 69:589–605, 2007.

[18] I. Fellows. Pseudo-optimal solutions to texas hold'em poker with improved chance node abstraction. 2010.

[19] S. G. Ficici and J. B. Pollack. A game-theoretic memory mechanism for coevolution. In *GECCO*, 2003.

[20] D. Fudenberg and D. M. Kreps. Learning mixed equilibria. *GEB*, 5:320–367, 1993.

[21] D. Fudenberg and D. K. Levine. Consistency and cautious fictitious play. *JEDC*, 19:1065–1089, 1995.

[22] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. The MIT Press, 1998.

[23] T. Galla and J. Farmer. Complex dynamics in learning complicated games. *NAS*, 110(4):1232–1236, 2013.

[24] S. Hahn. The convergence of fictitious play in 3 x 3 games with strategic complementarities. *Economics Letters*, 64(1):57–60, 1999.

[25] J. Hofbauer. Stability for the best response dynamics. Preprint Vienna 1994. Revised Budapest, 1995.

[26] S. Jensen, D. Boley, M. Gini, and P. Schrater. Non-stationary policy learning in 2-player zero sum games. In *AAAI 20*, 2005.

[27] J. S. Jordan. Three problems in learning mixed-strategy nash equilibria. *GEB*, 5:368–386, 1993.

[28] V. Krishna. Learning in games with strategic complementarities. HBS Working Paper 92-073, 1992.

[29] D. S. Leslie. Convergent multiple-timescales reinforcement learning algorithms in normal form games. *AAP*, 13:1231–1251, 2003.

[30] P. Milgrom and J. Roberts. Rationalizability, learning, and equilibrium in games with strategic complementarities. *Econometrica*, 58:1255–1277, 1990.

[31] K. Miyasawa. On the convergence of the learning process in a 2 x 2 non-zero-sum two-person game. Technical report, Princeton University ERP, 1961.

[32] D. Monderer and L. S. Shapley. Fictitious play property for games with identical interests. *JET*, 68(1):258–265, 1996.

[33] J. F. Nash. Equilibrium points in n-person games. In *NAS*, 1950.

[34] J. L. Ny. On some extensions of fictitious play. 2006.

[35] J. Robinson. An iterative method of solving a game. *AOM*, 54:296–301, 1951.

[36] J. B. Sanders, T. Galla, and J. L. Shapiro. Effects of noise on convergent game-learning dynamics. *J. Phys. A*, 45(10):105001, 2012.

[37] L. Shapley. Some topics in two-person games. *Advances in Game Theory*, 3:1–28, 1963.

[38] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *UAI 16*, 2000.

[39] M. Smyrnakis and D. S. Leslie. Dynamic opponent modelling in fictitious play. *CJ*, 53:1344–1359, 2010.

[40] M. Smyrnakis and D. S. Leslie. Adaptive forgetting factor fictitious play. *CoRR*, abs/1112.2315:1–23, 2011.

[41] R. Turner, Y. Saatci, and C. E. Rasmussen. Adaptive sequential bayesian change point detection. In *NIPS Temporal Segmentation Workshop*, 2009.

[42] C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, Cambridge, 1989.

[43] R. C. Wilson, M. R. Nassar, and J. I. Gold. Bayesian online learning of the hazard rate in change-point problems. *NC*, 22:2452–2476, 2010.

[44] C. Zhang and V. Lesser. Multi-agent learning with policy prediction. In *AAAI 24*, 2010.

APPENDIX

A. FICTITIOUS PLAY EXAMPLE DETAILS

We solve the system for one cycle (e.g. Figure 1b). Each arm of the diamond is a period of time when one agent is playing the correct strategy and the other is playing the incorrect one. At the end of the cycle the system is closer to the origin. The calculation works as follows. One calculates the time to traverse each of the diamond's four arms, and its vertex locations. The calculation is slightly complicated by the first step of each arm, where one strategy is updated by 0 instead of ± 1. The next theorem shows the calculation.

THEOREM 1. *Starting the dynamical system at time t_0 with $y(t_0) = y_0$ and $x(t_0) = 0$, and assuming that $y(t_0)$ is the time average of a series of ± 1 values (so that when it changes sign, it will go through the value 0 exactly), the time taken to traverse the first cycle is*

$$T_1 = 4t_0y_0 + 10, \tag{16}$$

and the value of y at the end of the first cycle is,

$$y(t_0 + T_1) = \frac{t_0y_0 + 4}{t_0(1 + 4y_0) + 10}. \tag{17}$$

PROOF. *A solution to recurrence relations (9) and (10) will take the form,*

$$a_i(t+\tau) = \frac{ta_i(t)}{t+\tau} \pm \begin{cases} \frac{\tau-1}{t+\tau} & \text{if opponent } a_{-i}(t) = 0, \\ \frac{\tau}{t+\tau} & \text{otherwise.} \end{cases} \tag{18}$$

Here a_i is either x or y, a_{-i} is the alternative, and the sign is positive if a_i is increasing or negative if a_i is decreasing. The time period, τ, is between changes of strategy. Traversing an arm of the cycle (or a quarter cycle), starts with $a_i = 0$ and finishes when $a_{-i} = 0$. The time taken, and values of a_i and a_{-i} after this time will be

$$\tau = ta_i + 1, \tag{19}$$

$$a_{-i} = \pm\frac{ta_i + 1}{t + ta_i + 1} \tag{20}$$

$$a_i = 0. \tag{21}$$

Here t is the time at the start of the traversal of this arm. To get the properties of the cycle, we just have to iterate this four times starting at time t_0, with $x(t_0) = 0$ and $y(t_0)$ positive, and alternate the roles of x and y. First, subtracting from y and adding to x until $y = 0$ (arm 1), then subtracting from y and x until $x = 0$ (arm 2), then adding to y and subtracting from x until $y = 0$ (arm 3), then adding to y and x until $x = 0$ which completes the cycle. Iterating Equations (19) and (20) four times gives the result. □

To verify the claims in Section 3.3, we use the following.

COROLLARY 1. *The period of the ith cycle is proportional to i, and the time after the ith cycle is $O(i^2)$.*

PROOF. *Define T_i as the period of the ith cycle, t_i as the time after the ith cycle, and y_i as the value of y after the ith cycle. The recurrence relations implied by Equations (16) and (17) are,*

$$T_i = 4t_{i-1}y_{i-1} + 10, \tag{22}$$

$$y(t_{i-1} + T_i) = y_i = \frac{t_{i-1}y_{i-1} + 4}{t_{i-1}(1 + 4y_{i-1}) + 10} \tag{23}$$

Using $t_{i-1} = t_{i-2} + T_{i-1}$ and the value for y_{i-1} from Equation (23) yields the recursion relation

$$T_i = 4(t_{i-2}(1+4y_{i-2})+10)\frac{t_{i-2}y_{i-2} + 4}{t_{i-2}(1 + 4y_{i-2} + 10} = T_{i-1}+16. \tag{24}$$

This is solved as $T_i = T_{i-1} + 16(i-1)$. From an asymptotic perspective, this proves the result. The time after i cycles is

$$t_i = t_0 + \sum_{j=1}^{i} T_j = t_0 + iT_1 + 16\sum_{j=1}^{i-1} j = t_0 + iT_1 + 16\frac{i(i-1)}{2}$$

$$= t_0 + i(T_1 - 8) + 8i^2 \text{ which is } O(i^2). \tag{25}$$

A starting point consistent with our assumptions is $t_0 = 1$, $y_0 = 1$, and $x_0 = 0$. Thus $T_1 = 4t_0y_0 + 10 = 14$, and

$$t_i = t_0 + i(T_1 - 8) + 8i^2 = 1 + 6i + 8i^2. \tag{26}$$

So, the cycle period grows like i, and the time between cycles grows like i^2. □

COROLLARY 2. *The system converges to 0 in inverse proportion to the number of cycles.*

PROOF. *The task is to show that y decreases like $1/i$. We need to solve recursion relation (23). It is helpful to see that t_i from Equation (26) (where $t_0 = 1$, $y_0 = 1$, and $x_0 = 0$) can be factorised as $(1 + 2i)(1 + 4i)$. We solve recursion relation (23) by ansatz, guessing that $y_i = 1/(1 + 2i)$. Due to the factorisation, $y_it_i = 1 + 4i$, which gives*

$$y_{i+1} = \frac{4i + 5}{(1 + 2i)(1 + 4i) + 4(1 + 4i) + 10} = \frac{4i + 5}{8i^2 + 22i + 15}$$

$$= \frac{1}{2i + 3} = \frac{1}{1 + 2(i + 1)} \tag{27}$$

So the ansatz works, and y shrinks per cycle like $\Theta(1/i)$. □

COROLLARY 3. *The system defined by recurrence relations (9) and (10) converges like inverse square-root of time.*

PROOF. *According to Corollary 2, the system gets closer to the fixed point inversely with the number of cycles, and due to Corollary 2, the time to complete i cycles scales like i^2. Thus, in time t^2 the system gets closer to the fixed point by $1/t$, so in time t it gets closer to the fixed point by $1/\sqrt{t}$.* □

B. MAXIMUM CONVERGENCE RATE OF AN EMPIRICAL PROBABILITY

We claim Equation (4) cannot converge faster than $1/t$.

PROOF.

1. For any $\alpha \in \mathcal{A}$, its empirical probability in Equation (4), $\sum_{i=1}^{t} [\![\alpha_i = \alpha]\!]/t$, cannot converge to any probability, $0 \le p \le 1$, faster than $S(t) = \text{nint}(tp)/t$ where nint is the nearest integer (or round) function.
2. If $f(t) = D(S(t)||p)$ where D is the divergence, then
   (a) $f(t) \le 0.5/t$, and
   (b) $f(t) > c/t$ infinitely often where $0 < c < 0.5$.

From point 2a it follows that $f(t) = O(1/t)$, and from point 2b it follows that $\nexists g(t) : g(t) = o(1/t), f(t) = O(g(t))$. □

C. PROOF OF NASH EQUILIBRIUM IN GENERALISED MATCHING PENNIES

We claim that the game in Table 2 has one mixed strategy Nash equilibrium at $(p = p*, q = q*)$.

PROOF. The expected reward to player 1 is $V_1 = pq\left(\frac{2}{q*} - 3\right) - p(1-q) - (1-p)q + (1-p)(1-q)$. The gradient of V_1 with respect to p is $\frac{\partial V_1}{\partial p} = q\left(\frac{2}{q*}\right) - 2$. Thus, if $q = q*$, then $\frac{\partial V_1}{\partial p} = 0$. The expected reward to player 2 is $V_2 = pq\left(-\frac{2}{p*} + 3\right) + p(1-q) + (1-p)q - (1-p)(1-q)$. The gradient of V_2 with respect to q is $\frac{\partial V_2}{\partial q} = p\left(-\frac{2}{p*}\right) + 2$. Thus, if $p = p*$, then $\frac{\partial V_2}{\partial q} = 0$. □