Extending Fictitious Play with Pattern Recognition

Ronald Chu and Gerard Vreeswijk

Department of Information and Computing Sciences,
Faculty of Science, Utrecht University, The Netherlands

Abstract. Fictitious play, an algorithm to predict the opponents next move based on the observed history of play, is one of the oldest simple yet very effective algorithms in game theory. Although using pattern recognition as a more sophisticated way to analyze the history of play seems a logical step, there is little research available on this subject. In this paper we will examine two different types of pattern recognition, and formulate several algorithms that incorporate these approaches. These algorithms and the basic fictitious play variants they extend are empirically tested in eight tournaments on some well known formal-form games. The results obtained will show that adding pattern recognition to fictitious play improves performance, and demonstrate the general possibilities of applying pattern recognition to agents in game theory.

1 Introduction

The field of game theory studies strategic decision making, where a fixed number of players have a limited number of possible actions to choose from. The combination of the actions of all players, their joint actions, determines the outcome of the game. The mathematician George W. Brown describes his fictitious play (FP) algorithm as follows[2]:

> *"The iterative method in question can be loosely characterized by the fact that it rests on the traditional statistician's philosophy of basing future decisions on the relevant history. Visualize two statisticians, perhaps ignorant of min-max theory, playing many plays of the same discrete zero-sum game. One might naturally expect a statistician to keep track of the opponent's past plays and, in the absence of a more sophisticated calculation, perhaps to choose at each play the optimum pure strategy against the mixture represented by all the opponent's past plays."*

Fictitious play uses the observations from the past to build an observed frequency distribution of the actions of its opponent(s) and uses the action(s) with the highest observed frequency as a prediction of the opponent(s) next move. It then chooses a strictly myopic response to maximize its expected payoff. It was created as a heuristic for computing Nash equilibria by playing a fictitious game against itself, and proven valid by Robbinson[11]. Note the term fictitious play may be misleading in its current use as a model used by an actual player, since there is nothing fictional about the observed history of play and the algorithm does not "play a fictional game in its head" to make its predictions.

A next logical step in the evolution of the fictitious play algorithm is to perform a more sophisticated analysis of the history of play. There are of course many different approaches one can take to perform such an analysis. In this paper we will focus on performing pattern recognition; looking at sequences of actions instead of single actions in the history of play. Although pattern recognition is a broad and established field with much ongoing research there exists, as Spiliopoulos puts it, "a conspicuous gap in the literature regarding learning in games – the absence of empirical verification of learning rules involving pattern recognition"[14].

In the next section will we give a thorough overview of the basic fictitious play algorithm and two common adaptations. We will then examine two different approaches to add pattern recognition, and formulate several algorithms that extend fictitious play with the proposed pattern recognition. Finally we will formulate a set of experiments to obtain empirical data on the performance of the pattern recognizing algorithms and the basic fictitious play algorithms they extend. An analysis of the data obtained will show if, how and why pattern recognition influences performance. An extensive analysis and schematic overview of the data obtained can be found in[4].

2 Fictitious play

2.1 Brown's original algorithm

A normal-form game in game theory is a description of a game in the form of all players' strategy spaces and payoff functions. A n-player game has a finite strategy space $X = \prod_i X_i$. The strategy space X_i for each individual player $i = 1, 2, \ldots, n$ consists of all actions available to player i. There is a payoff (or utility) function $u_i(x)$ for each player i which determines the utility received by player i given the joint actions $x \in X$ played that round.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{pmatrix} 1,1 & 0,0 \\ 0,0 & 2,2 \end{pmatrix} \end{array}$$

Fig. 2.1: A normal-form 2-player game in a typical payoff matrix representation

As mentioned before FP uses the observed history of play to determine its actions. Because the lack of history the first action x_i^1 is specified arbitrarily[1]. The algorithm consists of two components: a forecasting and a response component. The game is played several rounds, and for each time index (or round number) $t \in \mathbb{N}^0$ the n-tuple $x^t = (x_1^t, x_2^t, \ldots, x_n^t)$ where $x^t \in X$, consists of the actions played that round by all players. Let $h^t = (x^1, x^2, \ldots, x^t)$ be the observed history up to round t. The forecasting function $p_i^t(x_j)$, where $x_j \in X_j$, returns the proportion of the time when x_j was played in the history h^t:

$$p_i^t(x_j) = \frac{\sum\limits_{u=1}^{t} I^{t-u}(x_j)}{t} \tag{2.1}$$

Let $p^t = \prod_i p_i^t$ be the associated product distribution. The indicator function $I^t(x_j)$ returns 1 if $x_j = x_j^t$ (action x_j was played by player j at time t) and returns 0 otherwise. Note that $\sum_{x_i \in X_i} p_i^t(x_i) = 1$ for every i and every t.

The response is an action from the best-reply correspondence $BR_i(p^t)$; the set of all actions of i that maximize expected utility given p^t. The utility of a probability distribution is defined as the expected utility of its outcomes by defining $u_i(p) = \sum_{x \in X} u_i(x)p(x)$. Let Δ_i denote the set of all probability distributions on the set X_i, and let $\Delta = \prod \Delta_i$. Let $p_i^t \in \Delta_i$ and $p^t \in \Delta$ denote the observed probability distribution in round t for the actions of player i and the joint actions of all players, respectively. For every $p \in \Delta$ and every $x \in X$ the probability of x is $p^t(x) = \prod_i p_i^t(x_i)$. To be able to predict the expected utility for each action we need to be able to distinguish between our actions X_i and the actions of the other players $X_{-i} = \prod_{i \neq j} X_j$. In the same fashion let $\Delta_{-i} = \prod_{j \neq i} \Delta_j$ and let $p_{-i}^t \in \Delta_{-i}$. Now we can combine an action x_i and a partial probability distribution p_{-i} to obtain a full probability distribution $(x_i, p_{-i}) \in \Delta$ that places a probability 1 on x_i, and define the best-reply correspondence as:

$$BR_i(p_{-i}^t) = \{x_i \in X_i : u_i(x_i, p_{-i}^t) \geq u_i(x_i', p_{-i}^t) \text{ for all } x_i' \in X_i\}.$$

2.2 Adaptations

The original algorithm has a very rigid specification of predicting according to the empirical distribution of play and choosing an action to

[1] The superscript t in x^t indicates a time index, not an exponent. When exponentiation is intended the base is a number or between brackets

maximize the immediate expected utility. Fortunately there are possible modifications without losing the properties of convergence[6]. We will discuss two adaptations: smoothed and weighted fictitious play.

2.2.1 Smoothed fictitious play In smoothed fictitious play (SFP) the players' responses are smoothed by small trembles or random shocks. It was developed by Fudenberg and Kreps[6] along the lines of Harsanyi's purification theorem[8]. They define smooth fictitious play as the family of algorithms that employ a standard FP forecasting rule, and a response rule that maximizes the actual utility U_i:

$$U_i(x_i, p_{-i}) = u_i(x_i, p_{-i}) - \gamma_i w_i(x_i). \tag{2.2}$$

Here the smoothing function $w_i : \Delta_i \to \mathbb{R}$ is any smooth, differentiable and strictly concave function such that $|\nabla w_i(q_i)| \to \infty$ whenever x_i approaches the boundary of Δ_i, and the response parameter $\gamma_i > 0$.

To understand the variant of SFP we will be using, suppose the choice probabilities of player i are described by the logistic function[2] $q_i(x_i|p_{-i})$ which denotes the probability that action x_i will be played next:

$$q_i(x_i|p_{-i}) = \frac{e^{u_i(x_i,p_{-i})/\gamma_i}}{\sum\limits_{x'_i \in X_i} e^{u_i(x'_i,p_{-i})/\gamma_i}}. \tag{2.3}$$

If the response parameter γ_i is close to zero this function closely approximates the best-reply correspondence, while with a larger γ the probabilities are more evenly spread over all actions[3]. From an observer's standpoint a player using eq. 2.3 may look like he is using a best-reply function which is perturbed by small utility trembles; each period the actual utility $U_i = u_i + \epsilon_i^t$ is perturbated by the extreme-valued variable ϵ_i^t (whose cumulative distribution is $\ln P(\epsilon_i^t \leq z) = -e^{-z/\gamma_i}$). Another argument for using the logistic function involves the Shannon entropy. The amount of information conveyed by a probability distribution q_i can be represented by the entropy function $q_i \ln(q_i)$:

$$-\sum_{x'_i \in X_i} q_i(x'_i, p_{-i}) \ln(q_i(x'_i, p_{-i})).$$

[2] See McKelvey and Palfrey[10] for details about the quantal response equilibrium. The general idea is that players make errors, but since the probability of an action being chosen is related to its utility it is unlikely that costly errors are made.

[3] When $\gamma \to \infty$ the function $q_i(x_i|p_{-i}) \to \frac{1}{|X_i|}$, thus all actions have an equal chance of being chosen next.

It can be shown[7] that the optimal q_i is given by the logistic function if we define the actual utility (eq. 2.2) as a weighted combination of the utility u_i and the information gained from experimenting:

$$U_i(x_i, p_{-i}) = u_i(x_i, p_{-i}) - \gamma_i \sum_{x'_i \in X_i} q_i(x'_i, p_{-i}) \ln(q_i(x'_i, p_{-i})).$$

2.2.2 Weighted fictitious play The second adaptation of classic FP we will discuss involves techniques where the observations from the past decay over time, making them less influential than more recent observations. If, as classic fictitious play does, the entire history is considered it is harder to obtain change in beliefs and behavior as the history gets larger thus making it harder for fictitious play to catch up when the opponent changes its strategy. In weighted fictitious play (WFP), also known as exponential FP, a weight factor $0 \leq \gamma \leq 1$ is applied to the history[3]. It uses the original best-reply response rule and a modification of the forecasting rule (eq. 2.1), where every round every prior observation is multiplied with the weight factor (i.e. at time t an observation from $t' \leq t$ is multiplied with $(\gamma)^{t-t'}$):

$$p_i^t(x_j) = \frac{I^t(x_j) + \sum_{u=1}^{t-1} (\gamma)^u I^{t-u}(x_j)}{1 + \sum_{u=1}^{t-1} (\gamma)^u}. \tag{2.4}$$

The weight factor γ determines the rate of decay. When $\gamma = 1$ there is no decay and weighted fictitious play behaves like classic fictitious play. When $\gamma = 0$ the entire history decays instantly and weighted fictitious play behaves like the learning rule introduced by Cournot[5]

3 Pattern recognition

In this section we examine two distinct implementations of pattern recognition in fictitious play. Rothelli[12], Lahav[9] and Spiliopoulos[14,15] have created similar pattern recognizing algorithms to describe human learning behavior. Their models search the entire history of play for possible patterns that match the last few moves played, apply a form of decay and finally use the frequency distribution obtained to predict the next move. This approach is similar to conditional fictitious play proposed by Aoyagi[1], who proved that if two players both apply conditional fictitious play that recognizes patterns of the same length their beliefs converge to the equilibrium in zero-sum games with a unique Nash equilibrium. We will examine Spiliopoulos' model because it is an intuitive extension of weighted fictitious play and in its simplest form does not apply any additional transformations designed to model how humans form beliefs and make decisions.

Sonsino[13] proposes confused learning with cycle detection at the end of the realized history of play and proves convergence to a fixed pattern of pure Nash equilibria for a large class of games. We will briefly touch confused learning and focus on the cycle detection he proposes.

3.1 N-Period fictitious play

Spiliopoulos[14,15] proposes n-period fictitious play (FPN), where $n \in \mathbb{N}^+$, as an extension to weighted fictitious play which keeps track of sequences of actions of length n in the observed history of play.

It does not directly search for patterns, but uses the observations of the last $n-1$ rounds as a premise and uses only the part of the history when it forecasts the opponent's next move[4]. It does respond to patterns as they will be visible in the history; with $n = 3$ and a pattern AAB the probability $p(B|AA)$, for example, will be high. Even patterns longer than n can indirectly be derived from the history sometimes. The pattern $AABBA$ will result in higher probabilities $p(B|AB)$, $p(A|BB)$ and $p(A|BA)$, but both $p(A|AA)$ and $p(B|AA)$ will be about equal.

FP1 is exactly the same as WFP, and FP3, for example, keeps track of the observed frequencies of all possible patterns of three, which enables a forecast of the next round given what was played in the last two rounds. First the indicator function I^t is extended to indicate wether a sequence of actions was played. Let $\bar{x}_j$ be a sequence of n actions. The indicator $I^t(\bar{x}_j)$ return 1 if the sequence resembles the actions played in rounds $t-n+1, \ldots, t-1, t$, and 0 otherwise. The forecasting function p_i^t in equation 2.4 is extended to return the proportion of the time the actions $\bar{x}_j$ were played:

$$p_i^t(\bar{x}_j) = \frac{I^t(\bar{x}_j) + \sum_{u=1}^{t-1} (\gamma)^u I^{t-u}(\bar{x}_j)}{1 + \sum_{u=1}^{t-1} (\gamma)^u}$$

[4] This is similar to Aoyagi's conditional history. An important difference is that conditional FP uses the joint actions of both players as a premise, while we only consider the actions of the opponent.

which can then be used to calculate the proportion of the time the action x_j was played given that the actions $\bar{x}_j$ were played in the previous $n-1$ rounds:

$$p_i^t(x_j|\bar{x}_j) = \frac{p_i^t(\bar{x}_j + x_j)}{\sum\limits_{x'_j \in X_j} p_i^t(\bar{x}_j + x'_j)}.$$

Spiliopoulos continues by extending the FPN algorithm to allow the players' perceptions to follow commonly ascribed psychophysics principles, which we will not discuss here because human leaning behavior is beyond the scope of our investigation. We will use the n-period fictitious play algorithm in its simplest form in our experiment.

3.2 Cyclic pattern detection

Sonsino[13] proposes superimposing the ability of strategic pattern recognition of cyclic patterns that repeat successively at the observed path of play on a learning model called confused learning[5]. Detection is restricted to basic patterns; patterns that do not contain a smaller subpattern[6]. If a player recognizes such a repeated pattern it will assume the other players will continue to follow that pattern and, being strictly myopic, plays a best-reply response against the predicted next joint actions of the pattern.

For every pattern p with length l_p there is a uniform bound T_p such that the model must detect p with probability 1 if it has appeared almost T_p times (if the next round may complete T_p uninterrupted occurrences of p). To eliminate the possible inconsistency that the model detects more than one pattern with probability one at the same time, it is assumed that only patterns with length shorter than some fixed bound L are detected and that $T_p l_p \geq 3L$. This ensures T_p is large enough relative to L. The detection of patterns after a history-independent fixed number of repetitions is to stylized to represent any realistic learning effort. To accommodate any form of non-stationary pattern detection a set of necessary conditions is added to the model. Patterns longer than 2 must appear almost two times (again, if the next round may complete two uninterrupted occurrences of the pattern) in the past $2l_p - 1$ rounds. Patterns of length 2 must appear almost 2½ times in the past 4 rounds and patterns of length 1 must appear 2 times in the past 2 rounds. For the pattern $ABCD$, for example, the last 7 observations must be $ABCDABC$, while a shorter pattern AB requires the last 4 observations to be $BABA$.

[5] In confused learning players learn to choose the best learning model and are confused in the sense that they do not a priori know what the best strategy is.

[6] A and AAB are examples of basic patterns, $ABAB$ and AA are not

A detected pattern p remains detected until it is contradicted, and a pattern p may only be detected l_p rounds after the previous pattern was detected. The existence of these minimal and the necessary conditions define a range in which pattern detection can occur. Sonsino shows that convergence of behavior to any fixed strategy that is not a pure equilibrium is incompatible with pattern recognition[7], and that convergence of behavior to some mixed strategy profile is only possible if agents consider arbitrarily long histories (and thus impossible for agents with bounded memory).

4 Experiment Setup

The algorithms used in our experiment (listed in Table 1) are defined by their forecasting and response components so we can quickly inspect not only the performance of the algorithm as a whole but also the performance of the individual components without diving into the internal beliefs. This enables us to quickly identify which behavior to examine in more detail.

Model	**Name**	**Forecast**	**Response**	**Parameters**
Brown's original FP	FP	Simple	Best reply	
Smooth FP	SFP	Simple	Smoothed	$\gamma = 1$
Weighted FP	WFP	Weighted	Best reply	
Smooth weighted FP	SWFP	Weighted	Smoothed	$\gamma = 1$
N-pattern	FPN	N-Pattern	Best reply	$N = 2, 3$
Smoothed n-pattern	SFPN	N-Pattern	Smoothed	$N = 2, 3$, $\gamma = 1$
Weak cycle	FPwCL	Weak cycle	Best reply	$L = 2, 3, 20$
Smoothed weak cycle	SFPwCL	Weak cycle	Smoothed	$L = 2, 3, 20$, $\gamma = 1$
Strong cycle	FPsCL	Strong cycle	Best reply	$L = 2, 3, 20$
Smoothed strong cycle	SFPsCL	Strong cycle	Smoothed	$L = 2, 3, 20$, $\gamma = 1$

Table 1: FP models used in the experiment.

Simple forecasting (section 2.1)
Brown's original forecasting algorithm.
Weighted forecasting (section 2.2.2)
The weighted forecasting algorithm with a weight factor γ of 0.9.
N-Pattern forecasting (section 3.1)
The pattern detecting adaptation of WFP proposed by Spiliopoulos.

[7] A detected pattern can only sustain itself if it consists only of pure Nash equilibria, because the player will break other patterns by playing a best-reply action.

We will use a pattern length n of both 2 and 3, resulting in four different algorithms: FP2, SFP2, FP3 and SFP3. The weight factor γ is 0.9.

Weak cycle forecasting (section 3.2)
Patterns are only detected at the upper bound when the next round may complete T_p consecutive occurrences of the pattern. The pattern length L affect how quick a pattern is detected because $T_p = \text{floor}(3L/l_p)$. We will use the same short pattern length L of 2 and 3, and another large L of 20 resulting in six different algorithms. When a pattern is detected the algorithms returns a probability distribution where the next action in the pattern has probability of one. If no pattern is detected it employs the Simple forcasting algorithm.

Strong cycle forecasting (section 3.2)
Patterns are detected at the lower bound described by Sonsino. The pattern length L does not influence speed with which patterns are detected. We will use the same pattern lengths 2, 3 and 20, and this algorithms also employs the Simple forecasting algorithm when no pattern is detected.

Best reply response (section 2.1)
The best reply correspondence. If there is a tie (i.e. the set *BR* contains more than one action) then one is chosen at random.

Smoothed response (section 2.2.1)
The trembled best reply algorithm with a response parameter γ of 1.

There are 40 algorithms for 2×2 games and 60 for 3×3 games in our experiment[8]. For each game each algorithms faces all algorithms, themselves included, in a tournament where each game lasts 1000 rounds and is repeated 100 times. Finally they are tested with 100 randomly generated games, one tournament per game. The games used are:

Asymmetric Coordination game
There are two pure strategy equilibria and a mixed strategy equilibrium. Both players prefer the same equilibrium which Pareto dominates the other.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 5,5 & 0,0 \\ 0,0 & 3,3 \end{pmatrix} \end{array}$$

[8] There are 10 forecasters (FP, WFP, FP2, FP3, FPwC-2, FPwC-3, FPsC-2, FPsC-3 and FPsC-20) times 2 responders (Best reply, Smoothed) times 2 or 3 possible initial moves equals 40 or 60 algorithms.

Symmetric Coordination game
There are two pure strategy equilibria and a mixed strategy equilibrium. Neither equilibrium is preferred or dominated.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 1,1 & 0,0 \\ 0,0 & 1,1 \end{pmatrix} \end{array}$$

Shapley's game
The only equilibrium is a mixed strategy where each player plays each strategy with equal probability. The game is a non-zero sum variant of rock-paper-scissors designed to cause cyclic behavior in fictitious play algorithms.

$$\begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 1,0 & 0,0 & 0,1 \\ 0,1 & 1,0 & 0,0 \\ 0,0 & 0,1 & 1,0 \end{pmatrix} \end{array}$$

Battle of the sexes
There are two Pareto optimal pure strategy equilibria and a mixed strategy equilibrium. A different pure strategy equilibrium is preferred by each player.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 3,2 & 0,0 \\ 0,0 & 2,3 \end{pmatrix} \end{array}$$

Chicken
A variant of Battle of the Sexes with the same equilibria, where both players' actions corresponding to their preferred pure equilibria yields the worst outcome for both.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 0,0 & -1,1 \\ 1,-1 & -10,-10 \end{pmatrix} \end{array}$$

Matching pennies
Zero-sum game where the only equilibrium is a mixed strategy where each player plays each strategy with equal probability.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} -1,1 & 1,-1 \\ 1,-1 & -1,1 \end{pmatrix} \end{array}$$

Prisoner's dilemma
Each player has a dominant strategy resulting in a pure strategy equilibrium. The pure strategy joint actions A, A is Pareto optimal.

$$\begin{array}{c} \\ A \\ B \end{array}\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 2,2 & 0,3 \\ 3,0 & 1,1 \end{pmatrix} \end{array}$$

Random games
We've generated 100 random 2×2 games using GAMUT. All eight payoffs lie between 0 and 100.

5 Pattern Recognition Results

We will compare FPC with FP and FPN with WFP to see the difference with the basic fictitious play variants they are based on. All observations are based on an in-depth analysis of the results per game which can be found in the full thesis.

5.1 Cyclic pattern recognition

Both strong and weak pattern recognition perform better than FP in the random games, where only FPN scores higher. Exception is FPwC-20 which performs only marginally better than FP. This is to be expected since FPwC-20 detects patterns very slowly, in $3L = 60$ rounds, and thus falls back to FP very often. FPC scores higher than FP in Shapley's game and matching pennies, there is no performance improvement in the rest of the games. The behavior is exactly the same as FP in both coordination games and the prisoner's dilemma, almost the same with no clear advantage or disadvantage in chicken, and slightly disadvantageous in the battle of the sexes. FPsC performs better than FPwC in matching pennies and Shapley's game, there is no clear performance difference between strong and weak pattern recognition in all other games.

The pattern length L does not influence the performance of strong pattern recognition. Shorter patterns are allowed to be detected sooner, and with only two actions $\{A, B\}$ a pattern longer than three actions always contains either ABAB or AA making it impossible to detect longer patterns. Detection speed is independent of L and the ability to detect longer patterns is not an advantage because the detection of shorter pattern always blocks the detection of longer pattern.

The difference between weak pattern recognition with a different L responsible for the observed performance differences is simply that FPwC-3 is slightly slower in detecting patterns because of the higher pattern length L. In comparison with FPwC-2 and FPwC-3, FPwC-20 is much slower in detecting patterns and very likely to not detect any patterns at all because patterns need $^{60}/_{l_p}$ repetitions to be detected; short patterns are likely to be broken before they can be detected and long patterns are unlikely to occur at all. FPwC-20 performs exactly like FP in matching pennies and Shapley's game, and better than FP but still worse than FPwC-2 and FPwC-3 in the battle of the sexes, matching pennies, Shapley's game and in the random games.

5.2 N-Period fictitious play

In the random generated games, Shapley's game and matching pennies FPN outperforms all other algorithms tested in our experiment. It performs better than WFP in the symmetric coordination game. In the battle of the sexes WFP scores higher than FP3 but lower than FP2, and in the asymmetric coordination game WFP scores equal to FP3 and lower than FP2.

FPN quickly detects change if the opponent switches from one singleton pattern to another. Because AA and BB will occur more often than AB and BA in the observed history of play FPN will forecast the same action the opponent played in the previous round. This forecast is only wrong for one round when the opponent will switch actions.

The differences between $N = 2$ and $N = 3$ are marginal. FP3 performs slightly better in the random games and asymmetric coordination game, where FP2's forecast is wrong in one round. FP2 performs slightly better in matching pennies. There is no difference in performance between FP2 and FP3 in the remaining games.

6 Conclusion

In this paper we have studied two distictly different approaches to perform pattern recognition on the observerd history of play, obtained empirical data on the performance of those algorithms and the fictitious play variants they extend, and found the new pattern-recognition based FP variants to be significantly more effective than the traditional FP variants.

N-period fictitious play is an extension of weighted fictitious play inspired by pattern recognition, which has proved itself to be significantly more effective than WFP. FP3 performs only slightly better than FP2 in two, and slightly worse in one of the games tested in our experiment. We cannot at this stage, however, recommend not to use a pattern length $N > 2$. The costs are higher, but the effectiveness of a higher pattern length against other opponents, which are not related to FP, remains to be tested.

In contrast to FPN, the strong and weak pattern recognition capabilities of PFC have no relation at all to the fictitious play forecaster algorithm, which in our experiments only serves as a fallback forecaster in case no patterns are detected. It can easily be replaced by any other adaptive algorithm, as Sonsino describes. Furthermore, the strong and weak detection algorithms are merely implementations of the lower and upper bounds of the area where, in Sonsono's model, pattern recognition *may* occur. This does not mean that they are bad algorithms per se, nor does it mean that good or bad results necessarily imply cyclic pattern recognition at the end of the observed history of play is a good or bad strategy. The results of these two approaches do, however, give valuable insights that should be taken into account when designing a cyclic pattern recognizing algorithm.

The results we obtained are encouraging and illustrate the possibilities and merits of applying pattern recognition in machine learning and game theory. Adding a layer of pattern recognition on top of FP improves performance significantly in comparison to FP on its own and FPN nearly always outperforms WFP.

Fictitious play is a basic algorithm which has its limitations and weaknesses yet is very effective for its simplicity. The pattern recognition algorithms we studied share some of those strengths and weaknesses because they are either and extension of FP or use FP as a fallback strategy, and the matches show similar behavior because there were only FP-based opponents participating in the tournament. But FPN and FPC do show different behavior, overcoming FP's vulnerably to cyclic behavior and introducing new problems coordinating in self-play, for example.

The field of pattern recognition, however, is vast and offers many idea's and possibilities for (advanced) pattern recognition that have no relation at all to fictitious play. Confining further research to fictitious play is an unnecessary limitation on the broad spectrum of possibilities that pattern recognition has to offer. Despite the effectiveness of the pattern-detection based fictitious play algorithms studied, pattern detection should be seen as a stand-alone approach to machine learning. The best reply response works well in combination with the tested pattern detecting forecasts, but we should not ignore possibilities where pattern recognition provides both components or a complete learning algorithm where such a distinction is not applicable at all.

References

1. M. Aoyagi. Evolution of beliefs and the nash equilibrium of normal form games. *Journal of Economic Theory*, 70(2):444–469, 1996.
2. G.W. Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.
3. Y.W. Cheung and D. Friedman. Individual learning in normal form games: Some laboratory results. *Games and Economic Behavior*, 19(1):46–76, 1997.

4. R. Chu. Extending Fictitious Play with Pattern Recognition. Master's thesis, Universiteit Utrecht, 2013.
5. A. Cournot. Research into the mathematical principles of the theory of wealth (1838), trans. *BACON. New York*, 2, 1897.
6. D. Fudenberg and D. Kreps. *Learning mixed equilibria.* MIT Press, 1992.
7. D. Fudenberg and D.K. Levine. *The theory of learning in games*, volume 2. The MIT press, 1998.
8. J.C. Harsanyi. Games with randomly disturbed payoffs: A new rationale for mixed-strategy equilibrium points. *International Journal of Game Theory*, 2(1):1–23, 1973.
9. Y. Lahav. Behavioral pattern learning models for decision making in games. *Journal of Pattern Recognition Research*, 4(1):133–151, 2009.
10. R.D. McKelvey and T.R. Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.
11. J. Robinson. An iterative method of solving a game. *The Annals of Mathematics*, 54(2):296–301, 1951.
12. T.F. Rötheli. Pattern recognition and procedurally rational expectations. *Journal of economic behavior & organization*, 37(1):71–90, 1998.
13. D. Sonsino. Learning to learn, pattern recognition, and nash equilibrium. *Games and Economic Behavior*, 18(2):286–331, 1997.
14. L. Spiliopoulos. Pattern recognition and subjective belief learning in repeated mixed strategy games. 2009.
15. L. Spiliopoulos. Pattern recognition and subjective belief learning in a repeated constant-sum game. *Games and Economic Behavior*, 2012.
16. H. Peyton Young. Learning. In *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*, chapter 2, pages 25–43. Princeton University Press, 1998.