Evaluation, Simulation and Comparison of iterative learning algorithms **Fictitious Play**(FP) and **Smooth/Stochastic Fictitious Play**(s-FP)

ANSHUL VERMA
#1004730703

Abstract: This is a project paper submitted to Prof. Lacra Pavel. The project is a final project for the course ECE1657. The project aimed at the study of Fictitious Play and Smooth Fictitious Play learning algorithms for finite action games. The paper summarizes the major simulation result obtained from simulation and study. Using the simulation results a comparison between the two algorithms have been made. The paper also includes a section elaborating the effect of noise level in the sFP algorithm.

Introduction

Fictitious Play was one of the first iterative learning algorithm introduced by G.W. Brown [3] in 1951. The original learning scheme suggested and introduced by Brown is quite different from the 'fictitious play' algorithm used and talked about today [2]. In the original algorithm, Brown described the players updating their actions alternatively. Whereas the modern and more common version of the algorithm involves a simultaneous update of player actions. This approach is easier to study since it treats the players symmetrically.

In the fictitious play algorithm, players are bound to play their pure strategy best response myopic to their beliefs. The algorithm converges to a pure strategy Nash equilibria if it exists. The player's action for the algorithm can not converge to a mixed strategy Nash equilibria for finite iterations but in some cases where the actions of the player do not converge, their beliefs tend to converge to the mixed strategy Nash equilibria solution of other players. Thus for a game with no pure-strategy Nash equilibrium actions of the players will never converge to a mixed strategy Nash equilibria for pure fictitious play. But the empirical frequencies of each player's actions or their beliefs about other players can converge to the mixed strategy Nash equilibria solution.

Whereas, in smooth/stochastic fictitious play, players at each iteration are allowed to play mixed strategy best response to their belief about the other players. This best response is also called myopic perturbed best response because it depends on the belief of each player and is actually the best response of their perturbed costs. Since the players are allowed to play mixed strategies, actions of players for this algorithm can converge. If the algorithm converges after a sufficiently large number of iterations then the converging point of the game is considered to be the approximate Nash equilibria of the game (might be pure or mixed Nash equilibria).In fact, strict Nash equilibria of the perturbed game are the absorbing states for the smooth/stochastic fictitious play algorithm i.e. if the players play Nash-equilibrium of the perturbed game for one cycle then they continue playing the same strategy.

All the fictitious play algorithms assume the game to be stationary and players to be rational(i.e. them caring about their own cost only). The players are also assumed to know their costs and are able to see and adjust in the next iteration to what the other players do in the current iteration of the game. The fictitious play algorithms tends to respond slowly to any change in the cost matrix of the players with time. To improve this behaviour an alternate approach can be used to implement the fictitious play algorithm. In this alternate approach observed strategies of players are given weights and each player plays the best response to their weighted belief in the next iteration of the game. These beliefs arise from the weighted past behaviour of the other players and the original belief of the player about the other players.

In stochastic Fictitious play costs of all players are perturbed by a deterministic or a random noise, and each player plays the best response to their belief about other players using their perturbed cost matrix. It was observed that if the players are allowed to play the mixed strategy best response to their actual cost without any perturbation then their behaviour was similar to what it would be for Fictitious Play algorithm where the players are allowed to play only the pure strategy best response to their beliefs about the other players. To understand this behaviour, the algorithm's performance for different noise level given a deterministic perturbation is studied.

Along with the implementation of all these algorithms. Convergence rate, run-time and complexity of all the algorithms are studied and also the classes of games for which each algorithm converge to Nash equilibrium is used to compare the algorithms.

Convergence of Learning Algorithm

The fictitious play algorithm is used to approximate the optimal strategy for a game. These algorithms are sequential procedures that approximate the optimal strategy closer to the Nash equilibrium with increasing iterations before it actually converges to the Nash equilibria. Type of convergence studied for fictitious play algorithms are:

1 Consider a sequence of play $a(1), a(2), ...$ with $a(t) = \{a_1(\text{t}),\, a_2(\text{t}),\, ...,\, a_N(\text{t})\}$. We say that the sequence $\{a(t)\}$ converges to $\bar{a}$ if there exists T such that $a(t) = \bar{a}$ for all $t \geq T$. The action of each player for the algorithm converges to the Nash equilibria of the game(i.e. after a sufficiently large number of

iterations the players always play the strategy corresponding to Nash equilibria and they have no incentive to deviate from their strategy).

2 Alternative notation of convergence for fictitious play algorithms is convergence of empirical frequency/beliefs of each player. The strategy profile corresponding to the product of these converging empirical frequencies converges to a Nash Equilibrium of the game.

In standard or pure-fictitious play, actions of each player can only converge to a pure strategy Nash equilibria of the game. Since actions can only take pure strategy profiles in pure fictitious play, it can only converge to a pure action corresponding to a pure strategy Nash equilibria if it exists. Thus the actions of player for a pure strategy fictitious play cannot converge to a pure strategy profile for a game with no pure strategy Nash equilibria. However even for games where actions of players might not converge the beliefs about their opponent's action might converge. These beliefs of players can only converge to the Nash equilibrium solution of the game. But the actions of the player for pure fictitious play can only converge to a pure strategy Nash equilibria.

Whereas, in smooth/stochastic fictitious play players are allowed to play mixed strategy best response to their beliefs which is also called myopic best response of the players. Thus even for a game with no pure strategy Nash equilibria, actions of players and their beliefs can converge. But unlike standard fictitious play, the beliefs and actions of players converge to an approximate Nash equilibria which is actually the Nash equilibria of the perturbed game. Usually for a game if actions of the players converge for sFP then their beliefs also converge.

The literature available shows that fictitious play algorithm converges for the given classes of 2-player finite action game:

(i) zero sum game[14]

(ii) game solvable by iterative elimination of strictly dominated strategies[13]

(iii) potential game[12][11]

(iv) 2X2 game[10]

(v) 2×N game with generic payoff[1].

There are specific classes of games where the fictitious play algorithm need not converge[9], and there are also certain classes of games where the global convergence of the fictitious play algorithm holds. [1, 8, 12, 11, 10, 13, 7, 5, 4, 15]

Game setup

Game $\boldsymbol{G}$ is played repeatedly. The game is played between N players, with $\boldsymbol{I}$ representing set of all players.

Any i^{th} player has m_i finite possible number of actions and Ω_i represents the discrete finite action set of the i^{th} player. $\boldsymbol{\Omega}$ is the stacked notation for action of all the players playing the game $\boldsymbol{G}$. $\mathbf{J_i}$ (of appropriate dimension) represents the cost matrix of player i and is a mapping from the actions played in a game to a real number. $\mathbf{a}$ represents the vector of actions of a player with $\mathbf{a}_i$representing action of player i

Mixed strategies of player i is represented by $\mathbf{\Delta_i}$. The set $\mathbf{\Delta_i}$ contains all the legal and allowed mixed strategies of player i over his possible m_i actions. Best response map is denoted by $\mathbf{\Phi}$, best response of the i^{th} player $\mathbf{\Phi_i}$ is a map from $\mathbf{\Delta_{-i}}$ to $\mathbf{\Delta_i}$. The best response map gives the set of best response of a player to a specific move of their opponents.

$\{\boldsymbol{G}(\boldsymbol{I}, \mathbf{\Omega}, \mathbf{J_i})\}$:

$$\boldsymbol{I} := \{1,\, 2,\, ...,\, \mathrm{N}\ \}$$
$$\Omega_i := \{1, 2, ..., m_i\}$$
$$\mathbf{\Omega} := \Omega_1 \times \Omega_2 \times .... \times \Omega_N$$
$$\mathbf{J_i} : \mathbf{\Omega} \rightarrow \mathbb{R}.$$
$$\mathbf{\Delta_i} := \{p_i \in \mathbb{R}^{m_i} | \textstyle\sum_{j=1}^{m_i} p_{ij} = 1 \text{ and } p_{ij} \geq 0\ \forall j\}$$
$$\mathbf{\Phi_i} : \mathbf{\Delta_{-i}} \Rightarrow \mathbf{\Delta_i}$$
$$\mathbf{\Phi_i}(a_{-i}) := arg\ max_{a_i \in \Delta_i}\ J_i(a_i, a_{-i})$$

Standard Fictitious Play

Standard fictitious play restricts players to play their pure strategy best response to their beliefs at any iteration of the game. We define η_i^t as a mixed strategy in Δ_{-i}, which is a belief of player i about other players action at t^{th} iteration of the game. $\eta_i^t(a_{-i})\ \forall\ a_{-i} \in \Omega_{-i}$, represents the belief of player i at iteration t about others playing the pure strategy combination a_{-i} and η_i^0 represents the fictitious past or belief of the player. (i.e. η_i^0 is belief before the first iteration of the game)

So each player keeps track of actions of the other players and plays their pure strategy best response to their belief about the other players action at any iteration t. Player $i's$ action at any iteration t is represented by $s_i(t)$. We define a pure strategy best response of i^{th} player as Φ_i^p, the same way as the best response function but in this case, the range for a_i only being Ω_i.

$$\Phi_i^p : \Delta_{-i} \rightarrow \Omega_i$$
$$\Phi_i^p(a_{-i}) := arg\ max_{a_i \in \Omega_i}\ J_i(a_i, a_{-i})$$
$$s_i(t+1) \in \Phi_i^p(\eta_i^t)$$

Algorithm 1 Standard fictitious play

```
a_i, a_{-i} ← [ ] (# empty list for action of players)
b_i, b_{-i} ← [ ] (# empty list for belief of players)
sb_i ← shape(Ω_i)
sb_{-i} ← shape(Ω_{-i})
b_i.append(random(sb_i)) (# initial belief)
b_{-i}.append(random(sb_{-i}))
J_i ← input(CM) (# input for cost matrix)
J_{-i} ← input(CM)
i ← 1
if i ≤ itr then (# itr = number of iterations)
    a_i.append(BR_p(b_i, J_i)) (# append action)
    a_{-i}.append(BR_p(b_{-i}, J_{-i}))
    z_i ← (i/(i+1)) b_i[i − 1] + (1/(i+1)) a_{-i}[t]
    z_{-i} ← (i/(i+1)) b_{-i}[i − 1] + (1/(i+1)) a_i[t]
    b_i.append(z_i)) (# append belief)
    b_{-i}.append(z_{-i}))
    i ← i + 1.
close
        (# BR_P is pure-strategy Φ_i^P function)
```

In case of multiple pure strategy best responses(tie) for a belief about the other players, one of the multiple pure strategy best responses is randomly chosen by the player. Thus players always play a pure strategy which is the best response to their belief about their opponents action.

Based on the action of the players for the current iteration each player updates their beliefs for the next iteration.

$$s_i(t+1) \in \Phi_i^p(\eta_i^t)$$
$$\eta_i^{(t+1)} := \frac{1}{t+1}\sum_{j=0}^{t} s_{-i}(j)$$
$$\eta_i^{(t+1)} := \frac{t}{t+1}\eta_i^t + \frac{1}{t+1}s_{-i}(t)$$

Using these equations, players recursively update their beliefs about the other players action for the next iteration of the game.

Beliefs of a player can be seen as empirical frequencies of their opponents action, which is the basis for second type of convergence of the fictitious play algorithm defined above. Whereas a_i is the action of the i^{th} player which needs to converge for first type of convergence of the fictitious play algorithm.

Stochastic Fictitious Play

In stochastic fictitious play, at each iteration players are allowed to play a mixed strategy perturbed best response $\widetilde{BR}_i$ to their beliefs about the other players. Where $\widetilde{BR}_i$ is defined as

$$\widetilde{BR}_i = \widetilde{\Phi}_i : \Delta_{-i} \to \Delta_i$$
$$\widetilde{\Phi}_i(\eta_i^t) = arg\ max_{x_i \in \Delta_i}\ \tilde{J}_i(x_i, \eta_i^t).$$

The perturbed best response function of the players gives the choice probability function of the players

$$[\widetilde{\Phi}_i^{\epsilon}(\eta_i^y)]_j = \mathbb{P}(arg\ max_{x_i \in \Delta_i}\ \tilde{J}_i(x_i, \eta_i^t) = j).$$

$\tilde{J}_i$ represents the perturbed noise, J can have deterministic perturbation (v_i) depending on the actions of player i. A common approach is to use Gibbs entropy of a player's action at any iteration (x_i). With a parameter $\epsilon > 0$, this results in smooth/perturbed best response function $\widetilde{\Phi}_i : \Delta_{-i} \to \Delta_i$ of *logit* function function.

$$\tilde{J}_i(x_i, \eta_i^t) = J_i(x_i, \eta_i^t) + \epsilon v_i(x_i)$$
$$v_i(x_i) = \sum_{j \in \Omega_{-i}} x_{ij} log_e(x_{ij})$$
$$\mathbb{P}_j = [\widetilde{\Phi}_i(\eta_i^t)]_j = \frac{exp(J_i(e_{ij}, \eta_i^t)/\epsilon)}{\sum_{k \in \Omega_i} exp(J_i(e_{ik}, \eta_i^t)/\epsilon)}$$

Where the parameter ϵ defines the noise level. When ϵ approaches zero the *logit* probability choice function approaches uniform randomization and when ϵ approaches infinity it approaches unperturbed maximization.
or $\tilde{J}_i^{\epsilon}$ can have stochastic perturbation of the form

$$\tilde{J}_i^{\epsilon}(e_{ij}, \eta_i^t) = J_i(e_{ij}, \eta_i^t) + \epsilon_{ij}.$$

e_{ij} represents the j^{th} pure strategy of player i. In stochastic perturbation, the perturbations are suppose to be zero-mean and *iid*(independent and identically distributed). For the purpose of project, deterministic perturbation was added to the cost matrix and *logit* choice function was used to evaluate the choice probabilities of different actions of a player.
If the stochastic perturbation ϵ_{ij} is used and not the deterministic perturbation then one needs to calculate the approximate choice probability function which increases the complexity and run-time of the algorithm. Thus the use of deterministic perturbation was preferred for implementation.

Algorithm 2 Stochastic fictitious play

```
a_i, a_{-i} ← [ ] (# empty list for action of players)
b_i, b_{-i} ← [ ] (# empty list for belief of players)
sb_i ← shape(Ω_i)
sb_{-i} ← shape(Ω_i)
b_i.append(random(sb_i)) (# initial belief)
b_{-i}.append(random(sb_{-i})) (# initial belief)
J_i ← input(CM) (#input for cost matrix)
J_{-i} ← input(CM) (#input for cost matrix)
v_i ← random(1) (#initial random entropy)
v_{-i} ← random(1)
ϵ ← random(1)
i ← 1
if i ≤ itr then (# itr = number of iterations)
    J̃_i ← J_i + ϵv_i (# perturbed cost)
    J̃_{-i} ← J_{-i} + ϵv_{-i}
    a_i.append(BR_per_s(b_i, J̃_i)) (# append action)
    a_{-i}.append(BR_per_s(b_{-i}, J̃_{-i}))
    v_i ← entropy(a_i[i]) (# new entropy)
    v_{-i} ← entropy(a_{-i}[i])
    z_i ← i/(i+1) b_i[i − 1] + 1/(i+1) a_{-i}[i]
    z_{-i} ← i/(i+1) b_{-i}[i − 1] + 1/(i+1) a_i[i]
    b_i.append(z_i)) (# append belief)
    b_{-i}.append(z_{-i}))
    i ← i + 1.
close
```

(# BR_per_s gives mixed-strategy, $\widetilde{\Phi}_i$ function)

By Theorem 2.1 in [6]

$$[\widetilde{BR}_i(\eta_i^t)]_j = [\widetilde{BR}_i^{\epsilon}(\eta_i^t)]_j$$

for some admissible v_i. Mostly, v_i is related to the entropy so that we obtain a *logit* type probability choice function for the actions of player i. The used deterministic perturbation(i.e. Gibbs Energy) results in the probability choice function($[\widetilde{BR}_i]_j$) to be of the form of normalized exponential function. Beliefs of players are defined in a way similar to the standard fictitious play(i.e. η_i^t represents a mixed strategy in Δ_{-i}, which is the belief of player i about the other players at t^{th} iteration of the game).

The belief(η_i^t) updating rule for stochastic/smooth fictitious play is similar to standard fictitious play. It is just that the actions of each player in an iteration can be mixed. Thus we consider action, x_i to be a vector of probability (x_{ij}) of a player playing any pure action (e_{ij}). Thus everything in the algorithm remains the same as in standard fictitious play and instead of using Φ_i^P we use Φ_i^{per-s} the perturbed stochastic best rest response function.

The stochastic fictitious play always converges to the NE of the perturbed game which is an approximation of the NE of the original game. ϵ(Noise level) defines the extent of perturbation in the game and it also affects the convergence rate and goodness of approximation generated by sFP algorithm.

Simulation Results

In this section results for the classes of games for which the two algorithms are guaranteed convergence is presented

2-player 2×2 Game

1. A 2×2 game with atleast one pure strategy Nash equilibrium.

G_1 : Prisoner's Dilemma Game

P1 \ P2	Confess	Deny
Confess	1,1	0,10
Deny	10,0	5,5

Cost Matrix for G_1.

FP, Starting with a randomized initial beliefs for the 2-players. Checking convergence of the algorithm.

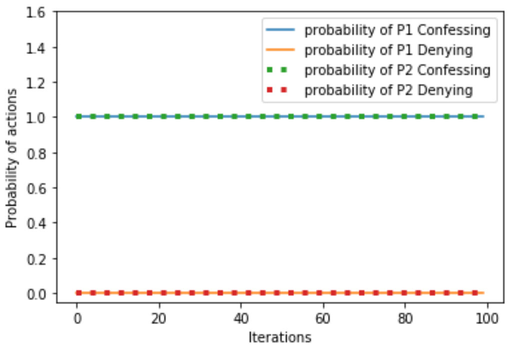

Figure 1: Action of Players for FP algorithm. (Type-1 convergence)

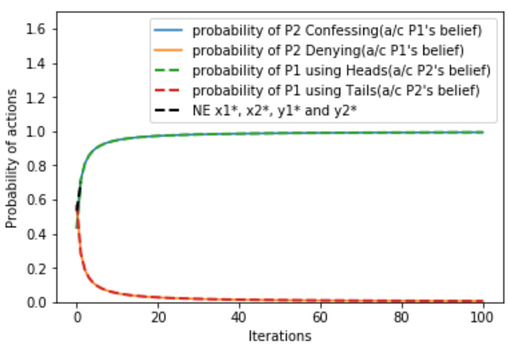

Figure 2: Empirical frequencies of Players for FP algorithm. (Type-2 convergence)

sFP, using the same random initial beliefs of players as used in FP algorithm. Checking convergence of the algorithm.

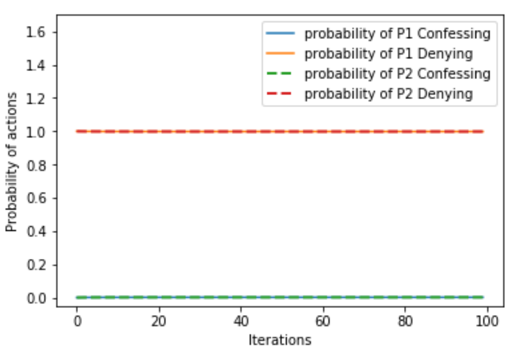

Figure 3: Action of Players for sFP algorithm. (Type-1 convergence)

From 1,2,3 and 4 both the FP and s-FP behave the same way for G1 and the actions and empirical frequencies of both the players converges to NE for both the algorithms. The convergence rate for the two algorithm is also very similar(≈40 iterations for empirical frequency and ≈1

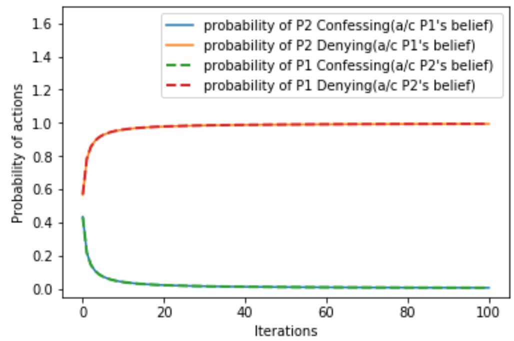

Figure 4: Empirical frequencies of Players for sFP algorithm. (Type-2 convergence)

iteration for action). In this game both the players have a dominating strategy which is to confess thus the game has a pure strategy $NE = (Confess, Confess)$.

2. A 2×2 game with no pure strategy Nash equilibrium.

G_2 : Matching Pennies Game

P1 \ P2	Head	Tail
Head	-1,1	1,-1
Tail	1,-1	-1,1

Cost Matrix for G_2.

FP, starting with some random initial beliefs of players. Checking convergence of the algorithm.

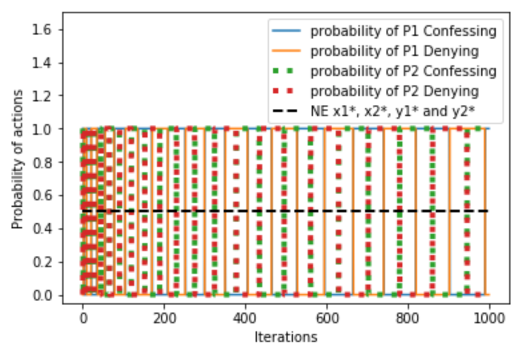

Figure 5: Action of Players for FP algorithm. (Type-1 Convergence)

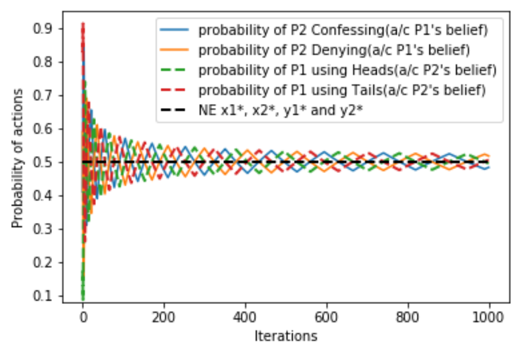

Figure 6: Empirical frequencies of Players for FP algorithm. (Type-2 Convergence)

In this game none of the player has a dominating strategy and there exists no pure strategy NE and the mixed strategy NE is $(x^*, y^*) = ([0.5, 0.5],[0.5, 0.5])$

sFP, using the same random initial beliefs of players as used in FP algorithm. Checking convergence of the algorithm.

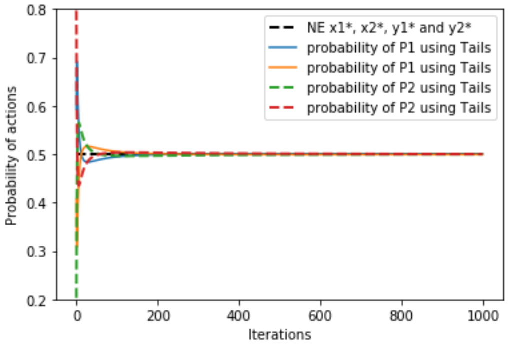

Figure 7: Actions of Players for sFP algorithm.

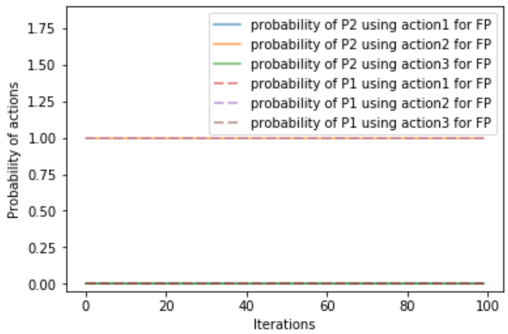

Figure 9: Action of Players for FP algorithm. (Type-1 convergence)

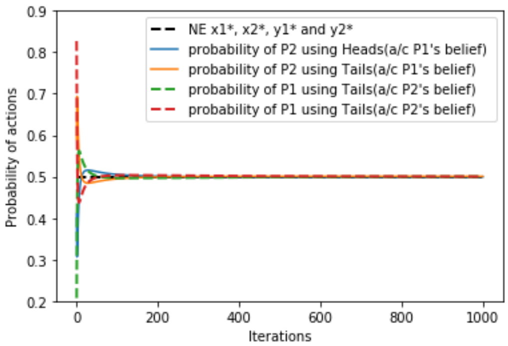

Figure 8: Empirical Frequencies of Players for sFP algorithm. (Type-2 Convergence)

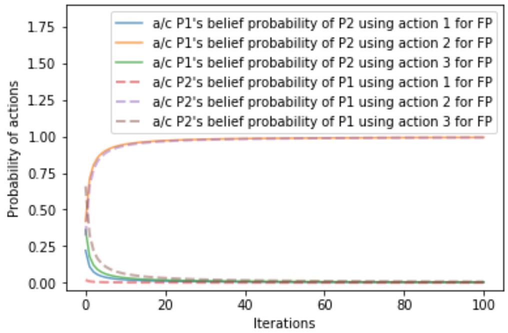

Figure 10: Empirical frequencies for FP algorithm. (Type-2 convergence)

From 5 it can be said that the actions of the players under FP do not converge for G2. Which was expected since there exists no pure strategy NE and 6 shows that even though the actions of players do not converge the empirical frequencies (i.e. beliefs) of the players converges and it converges to the mixed strategy NE of the game $x^* = [0.5, 0.5]$ and $y^* = [0.5, 0.5]$ represented by black line in 6.

From 7 actions of the players converge to mixed strategy NE for sFP algorithm. From 6 and 8 it can be seen that the convergence rate of the empirical frequencies of the player actions to NE is much faster for sFP algorithm than compare to FP algorithm. Empirical frequencies or beliefs of player almost converges after 200 iterations for sFP .Whereas, for the FP algorithm even after 1000 iterations empirical frequencies is around the NE but has still not completely converged to NE.

2-player zero-sum/constant-sum Game 3.

G_3 : 3×3 zero-sum Game

P2 / P1	C1	C2	C3
R1	5,-5	3,-3	3,-3
R2	1,-1	2,-2	0,0
R3	3,-3	4,-4	1,-1

Cost Matrix for G_3.

Considering the finite action 2-player game. A 3×3 zero sum game with a pure strategy NE.

FP, Starting with some random initial beliefs of the players. Checking convergence of the algorithm.

sFP, using the same random initial beliefs of as used in FP algorithm. Checking convergence of the algorithm.

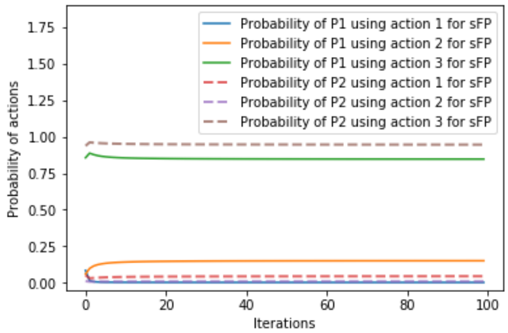

Figure 11: Action of Players for sFP algorithm. (Type-1 convergence)

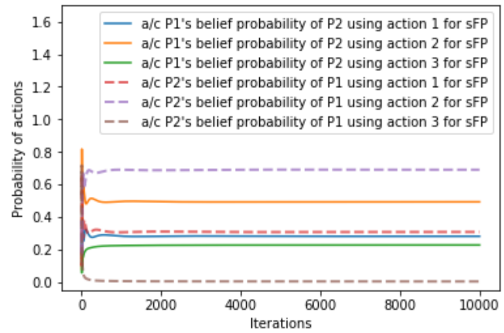

Figure 12: Empirical frequencies for sFP algorithm. (Type-2 convergence)

The game has a pure strategy NE which is $(x^*, y^*) = ([0, 1, 0], [0, 1, 0])$

From 9 it can be said that the actions of the players for FP converge for G3 and 10 shows that the beliefs of the players also converges to the the NE of the game. Beliefs of P1 and P2 has completely converged even after ≈ 20 iterations of G3.

From 11 it can be said that the actions of the players for sFP converge to approximate NE for G3. In 12, it can be seen that the beliefs of the players converges to approximate NE of the game in approximately 10 iterations of G3.

4. Considering a finite action 2-player game. A 3×3 zero sum game.

G_4 : 3×3 zero-sum Game

P2 / P1	C1	C2	C3
R1	0,0	8,-8	5,-5
R2	8,-8	4,-4	6,-6
R3	12,-12	-4,4	3,-3

Cost Matrix for G_4.

FP, starting with some initial random beliefs of player. Checking convergence of the algorithm.

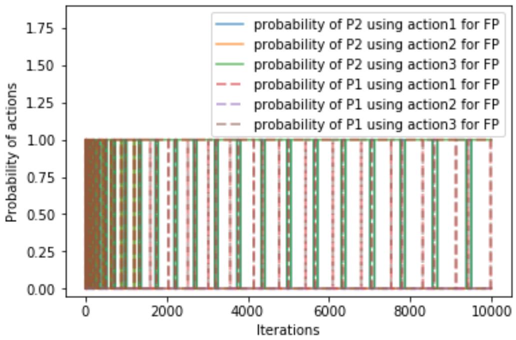

Figure 13: Action of Players for FP algorithm. (Type-1 convergence)

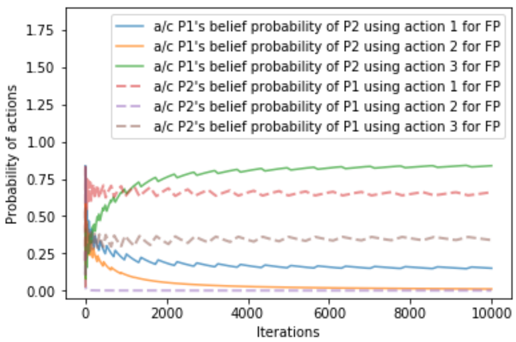

Figure 14: Empirical frequencies for FP algorithm. (Type-2 convergence)

sFP, using the same random initial beliefs of players as used in FP algorithm. Checking convergence of the algorithm.

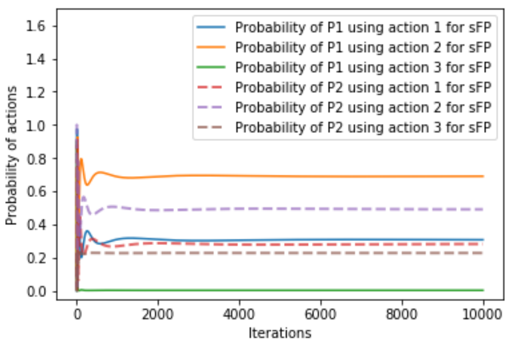

Figure 15: Action of Players for sFP algorithm. (Type-1 convergence)

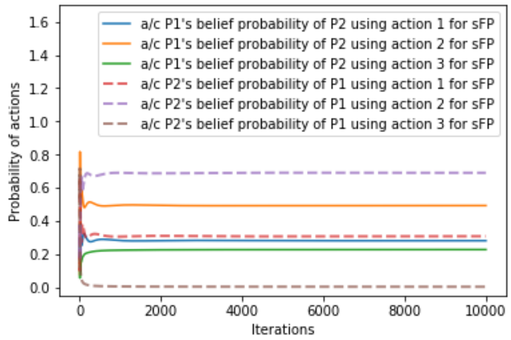

Figure 16: Empirical frequencies for sFP algorithm. (Type-2 convergence)

The game has no pure strategy NE and has a mixed strategy NE to which the FP algorithm converges. The game might have some other mixed strategy NE which the algorithms might converge to if we start with some other initial beliefs of the players.

13 shows that the actions of the players for FP do not converge for G4. From 14, it can be seen that the beliefs of the players are slowly converging to a fixed point which has to be the NE of the game since the FP algorithm is converging to that point. and beliefs of P1 and P2 have almost converged to the NE of the game after approximately 5000 iterations of G4. But the beliefs are still oscillating because players are not allowed to play mixed strategy.

15 and 16, show that the actions of the players and their beliefs converge to approximate NE of G4 for sFP. From 16 it can be seen that the beliefs of the players converges after approximately 500 iterations of G4.

2-player 2×N Generic-payoff Game

Def^n: *A generic game is one in which a small change(or non-affine transformation) of any one of the payoffs does not introduce new Nash equilibria or remove existing ones. In practice, this means that there should be no equalities between payoffs that are compared to determine a Nash equilibrium.*

G_5 : 2×4 Generic-payoff Game.

P2 / P1	C1	C2	C3	C4
R1	-10,0	-5,-1	4,2	6,-3
R2	-8,1	-2,0	6,-1	-8,0

Cost Matrix for G_5.

FP, starting with some random initial beliefs of players. Checking convergence of the algorithm.

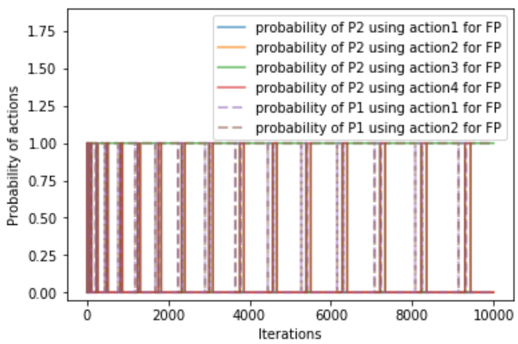

Figure 17: Action of Players for FP algorithm. (Type-1 convergence)

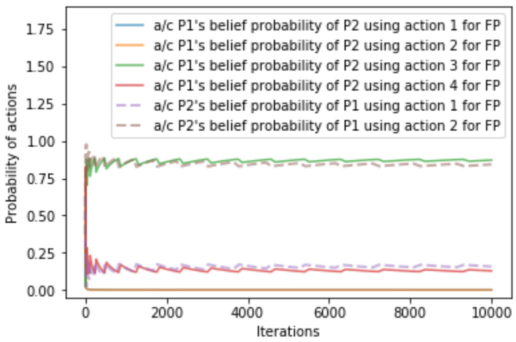

Figure 18: Empirical frequencies for FP algorithm. (Type-2 convergence)

sFP, using the same random initial beliefs of players as used in FP algorithm. Checking convergence of the algorithm.

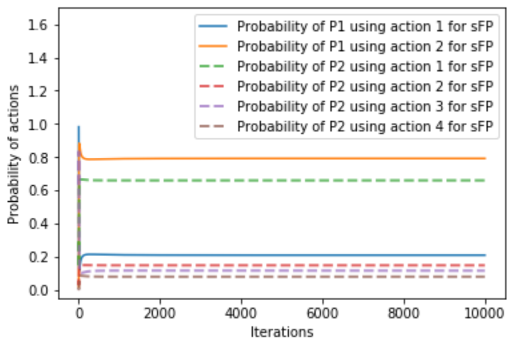

Figure 19: Action of Players for sFP algorithm. (Type-1 convergence)

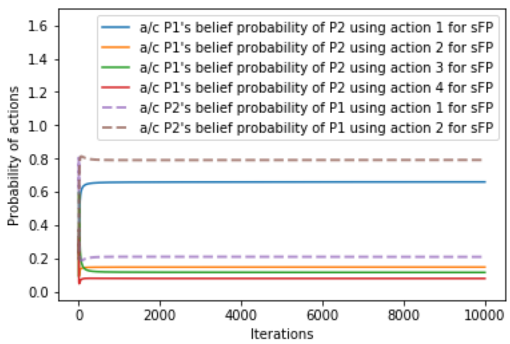

Figure 20: Empirical frequencies for sFP algorithm. (Type-2 convergence)

17 shows that the actions of the players for FP do not converge for G5. From 18 it can be said that the beliefs of players converge to the NE of the game after approximately 3000 iterations of the game but they still oscillate since the players are only allowed to play pure strategies in FP algorithm. The beliefs of the players are slowly converging to a fixed point which has to be the NE of the game since the FP algorithm is converging to that point.

19 shows that the actions of the players for sFP converge to approximate NE for G4. From 20, it can be said that the beliefs of the players converges to approximate NE of G4 after approximately 500 iterations of G4.

Algorithm Comparison

From all the example games considered for simulation it was observed that the other than the prisoner's dilemma game for all the other games the convergence rate of sFP was higher than the convergence rate of FP algorithm. sFP gives an approximate value of the Nash equilibrium of the game in much fewer iterations than FP. But the approximate value given by the sFP algorithm has a deviation from the actual NE of the game and this deviation is more if the convergence rate is higher in general.

Consider $m = max\{|\Omega_i|, \forall i \in I\}$ and $s = \sum_{i \in I} |\Omega_i|$. The order complexity of each iteration of sFP algorithm $\mathcal{O}$(sFP) depends on the $\widetilde{\Phi_i}$ function which includes evaluation of expected cost($\mathcal{O} \approx m$) of the players and *logit* function(choice probability function)($\mathcal{O} \approx m$) and the additional function used to find entropy of each player after each iteration($\mathcal{O} \approx s$).
$\mathcal{O}(\Phi_i^P) \approx 2 * m$(worst case) since it involves finding the expected cost of the players($\mathcal{O} \approx m$) and then finding the minimum($\mathcal{O} \approx m$) of all the expected costs.
Overall the complexity of $\mathcal{O}$(sFP) $>$ $\mathcal{O}$(FP) if both the algorithms are run for same number of iterations. Since, sFP converges faster than FP. sFP usually requires lesser number of iterations than the FP algorithm. Thus we multiply the number of iterations with $\mathcal{O}$(1-iteration) to find the $\mathcal{O}$ of n.

Run-time is a measure of complexity of the algorithm and for same number of iterations run-time of sFP algorithm was found to be higher than that of FP algorithm. The way the algorithms were implemented sFP required more memory than FP algorithm because of extra variable like entropy, perturbation and noise-level.

The classes of games for which the two algorithms have type-2 convergence is the same. But FP algorithms has type one convergence only if the game has atleast one NE. Thus the classes of games for which the two algorithms are guaranteed type-1 convergence is smaller for FP algorithm than for sFP algorithm.

Algorithms / Comparison measure	FP	s-FP
$\mathcal{O}$(iteration)	s+2m	2m
Run-time	T	$t < T$
Fixed point	actual NE	approximate NE
Convergence rate	c	$C > c$
Memory required	MEM	$mem < MEM$
Class of games offering type-1 convergence	S	$s \subset S$

Effect of noise on sFP's convergence

$\epsilon > 0$, in the deterministic perturbation defines the noise level. As the noise approaches zero in the the logit choice approaches unperturbed maximization and as ϵ tends to infinity it approaches uniform randomization. Using the approach for simulations setting $\epsilon = 0$ will give NaN since we are in the best response function there exists a division by ϵ, but analytically its easy to see that for limit $\epsilon = 0$, $\widetilde{J_i} = J_i$ and the system behaves as it is unperturbed, thus we get a really high convergence rate but the cost is almost unperturbed thus the convergence point of the sFP algorithm is close to the actual NE of the game. Whereas if we use a really high epsilon then the sFP converges almost instantly and the choice probability function tends towards the uniform randomization.

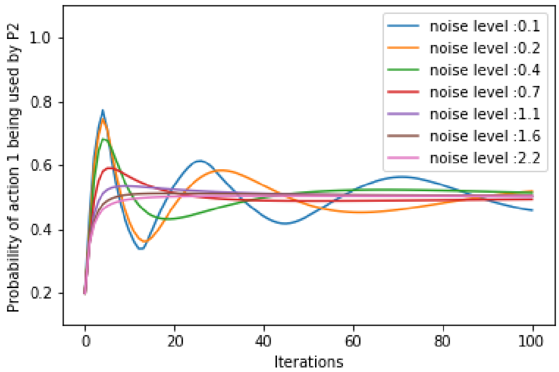

Figure 21: Low epsilon for MP(G2) game Probability of P2 using Heads.

From 21 its evident that the convergence rate increases as the value of ϵ increases.

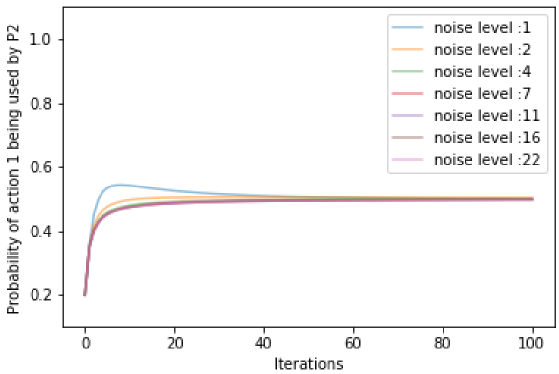

Figure 22: High epsilon for MP(G2) game Probability of P2 using Heads.

As the value of epsilon increases the *logit* choice function tends towards uniformly random probability of actions, but for G2 this behaviour of the choice function tending to random distribution has no effect on the convergence probability because the probability of P2 using heads when he behaves like a uniform RV is 0.5 which in this case is the NE of the game(only two strategies for each player).

For G2 only the convergence rate varies with ϵ. 21 shows that the increase in convergence rate is almost negligible after a sufficiently high ϵ.
Considering G4, to see the effect of randomization. Choosing P2's probability of using action 1 for the same initial conditions and different ϵ values.
23 confirms the variation of the convergence rate with epsilon and it also shows as the noise ϵ increases the game tends to move farther from the NE of the game and closer to the uniform randomization(0.333 in this case, since all the players have 3 strategies).

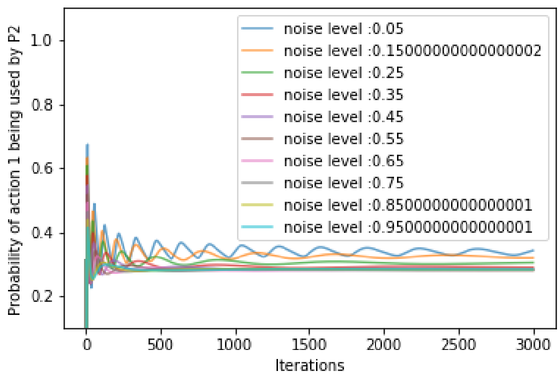

Figure 23: Low epsilon for G4 game Probability of P2 using Heads.

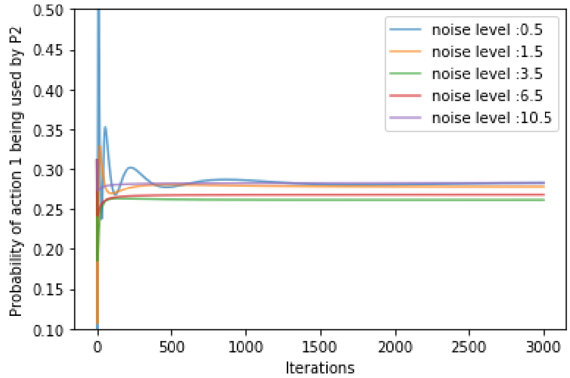

Figure 24: High epsilon for G4 game Probability of P2 using Heads.

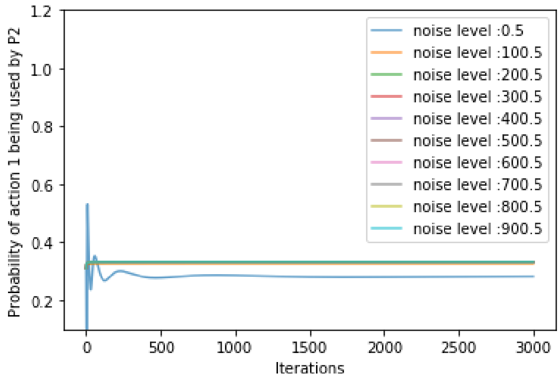

Figure 25: Extremely High epsilon for G4 showing randomization of Probability of P2.

24 and 25 confirms the fact that as statement stated before.

Summary

The fictitious play algorithm and the stochastic-fictitious play algorithms were simulated and observed for some classes of the games which guarantees convergence of these algorithms. The complexity of sFP was found to be larger than the complexity of the FP algorithm. The convergence rate of sFP was higher than the convergence rate of FP algorithm. But the FP algorithm tends to converge to the NE of the actual game whereas sFP only provides us with an approximate value of the NE and the goodness of the approximate depends on the noise-level. Both the algorithms have there pros and cons FP gives the NE of the exact game but at a lower convergence rate with complexity lesser than sFP. Whereas, sFP tends to give the approximate value of the NE of the game(actually NE of the perturbed game) at a higher convergence rate with more complexity than the FP algorithm. The rate of convergence depends on the noise level of the perturbation which also affects the deviation of the approximation of the NE from the actual NE of the game. The convergence rate of an algorithm is not the only measure of the goodness of an algorithm. Both the algorithms are useful, and FP is better than sFP if one desires to find the actual(precise) NE of the game. Whereas sFP is a better algorithm if one quickly wants to know the approximate NE of the game.

It was observed that the convergence rate and deviation from the actual NE of the game for stochastic-fictitious play have a trade-off. As ϵ increases the converging rate of the sFP algorithm increases but the deviation from the NE increases and the other-way around as ϵ decreases.

References

[1] Ulrich Berger. Fictitious play in 2× n games. Journal of Economic Theory, 120(2):139–154, 2005.

[2] Ulrich Berger. Brown's original fictitious play. Journal of Economic Theory, 135(1):572–578, 2007.

[3] George W Brown. Iterative solution of games by fictitious play. Activity analysis of production and allocation, 13(1):374–376, 1951.

[4] Ross Cressman and Christopher Ansell. Evolutionary dynamics and extensive form games, volume 5. MIT Press, 2003.

[5] Sunku Hahn. The convergence of fictitious play in 3× 3 games with strategic complementarities. Economics Letters, 64(1):57–60, 1999.

[6] Josef Hofbauer and William H Sandholm. On the global convergence of stochastic fictitious play. Econometrica, 70(6):2265–2294, 2002.

[7] Shlomit Hon-Snir, Dov Monderer, and Aner Sela. A learning approach to auctions. Journal of economic theory, 82(1):65–88, 1998.

[8] Vijay Krishna. Learning in games with strategic complementarities. Harvard Business School, 1992.

[9] Vijay Krishna and Tomas Sjöström. Learning in games: Fictitious play dynamics. In Cooperation: Game-Theoretic Approaches, pages 257–273. Springer, 1997.

[10] K. Miyazawa. On the convergence of the learning process in a 2×2 non-zero-sum two-person game,. Econometric Research Program, Princeton University, Research Memorandum No. 33, 1961.

[11] Dov Monderer and Lloyd S Shapley. Fictitious play property for games with identical interests. Journal of economic theory, 68(1):258–265, 1996.

[12] Dov Monderer and Lloyd S Shapley. Potential games. Games and economic behavior, 14(1):124–143, 1996.

[13] John H Nachbar. "evolutionary" selection dynamics in games: Convergence and limit properties. International journal of game theory, 19(1):59–89, 1990.

[14] Julia Robinson. An iterative method of solving a game. Annals of mathematics, pages 296–301, 1951.

[15] Aner Sela et al. Fictitious play in 2× 3 games. Games and Economic Behavior, 31(1):152–162, 2000.