ITERATIVE LEARNING ALGORITHMS
FICTITIOUS PLAY AND SMOOTH FICTITIOUS PLAY

Anshul Verma
1004730703

University of Toronto

ECE1657
Game Theory and Evolutionary Games

7^{th} December 2018

Content

Introdution
- Fictitious Play Algorithm
- Stochastic Fictitious Play Algorithm

Simulation Results

Comparison
- Order Complexity
- Comparison

Effect of noise in sFP
- Convergence rate
- Deviation

Summary

Fictitious Play algorithm was originally introduced by G.W. Brown in 1951. It was the first iterative learning scheme ever suggested.

Fictitious Play Algorithms

In fictitious play algorithms player's have some belief about their opponent's action and they try to maximize their payoff given their belief about the other players.
In these algorithms, players are assumed to know their cost matrix and they can observe the actions of each player and are allowed to update their belief simultaneously.

- The modern version of fictitious play algorithm is quite different from what was originally introduced by Brown.
- Each player update their action simultaneously.
- These algorithms assume players to behave rationally.
- Each player tries to play the best response to their belief about the other players.
- Each player update their belief after every iteration.

Standard Fictitious Play Algorithm

In this algorithm, the players have some belief about their opponent and they play pure strategy best response(BR_i^P) to their belief about the opponent's action.

Pure strategy best response BR_i^P:$\Delta_{-i} \rightarrow \Omega_i$ is defined as the pure strategy of player i that maximizes his expected cost given that others play the strategy that player i believes that they will play.

$$BR_i^P(x_{-i}) = \arg\ min_j J_i(e_{ij}, x_{-i})$$

- Each player starts with some random initial belief about other players.
- Since the players are only allowed to play pure-strategies their actions can only converge to pure-strategy profiles.
- If the actions of player converge in fictitious play algorithm then equilibrium point has to be pure strategy NE.
- The state sequence $\{x^t\}$ obtained by fictitious play is a non-stationary discrete-time Markov process.
- Empirical frequencies(beliefs) of players can also converge the mixed strategy Nash equilibrium of the game.

Algorithm 1 Standard fictitious play

a_i, $a_{-i} \leftarrow [\,]$ (# empty list for action of players)
b_i, $b_{-i} \leftarrow [\,]$ (# empty list for belief of players)
$sb_i \leftarrow shape(\Omega_i)$
$sb_{-i} \leftarrow shape(\Omega_{-i})$
b_i.append($random(sb_i)$) (# initial belief)
b_{-i}.append($random(sb_{-i})$)
$J_i \leftarrow input(CM)$ (# input for cost matrix)
$J_{-i} \leftarrow input(CM)$
$i \leftarrow 1$
if $i \leq itr$ **then** (# itr = number of iterations)
    $a_i.append(BR_p(b_i, J_i))$ (# append action)
    $a_{-i}.append(BR_p(b_{-i}, J_{-i}))$
    $z_i \leftarrow \frac{i}{i+1} b_i[i-1] + \frac{1}{i+1} a_{-i}[t]$
    $z_{-i} \leftarrow \frac{i}{i+1} b_{-i}[i-1] + \frac{1}{i+1} a_i[t]$
    $b_i.append(z_i)$) (# append belief)
    $b_{-i}.append(z_{-i})$)
    $i \leftarrow i + 1$.
close
    (# BR_P is pure-strategy Φ_i^P function)

Stochastic Fictitious Play Algorithm

In this algorithm players are allowed to play mixed strategies in general. Each player has some belief about the other players and they play their perturbed best response($\widetilde{BR}_i$) to their belief about their opponent's action.

Perturbed best response $\widetilde{BR}_i$: $\Delta_{-i} \to \Delta_i$, is defined as the BR to the perturbed cost of the player. The perturbation to the cost can be deterministic or random and the $\widetilde{BR}_i(x_{-i})$ gives the choice probability function with which the player will play a certain strategy.

$$\mathbb{P}[e_i^k = e_{ij} | x_{-i}] = [\widetilde{BR}_i(x_{-i})]_j$$
$$\widetilde{BR}_i(x_{-i}) = \arg\, min_{x_i \in \Delta_i} \widetilde{J}_i(x_i, x_{-i})$$
$$\widetilde{J}_i(x_i, x_{-i}) = J_i(x_i, x_{-i}) - \epsilon v_i(x_i)$$

- The perturbation can be of the deterministic of form $\widetilde{J}_i(x_i, x_{-i}) = J_i(x_i, x_{-i}) - \epsilon v_i(x_i)$ or random of form $\widetilde{J}_i^{\varepsilon}(e_{ij}, x_{-i}) = J_i(e_{ij}, x_{-i}) - \varepsilon_{ij}$.
- If deterministic perturbation is of the form of Gibbs Entropy i.e. $v_i(x_i) = \sum_{j \in \Omega_{-i}} x_{ij} log(x_{ij})$ then the choice probability (perturbed best response) is of *logit* form $[\widetilde{BR}_i(x_{-i})]_j = \frac{exp(\frac{1}{\epsilon} J_i(e_{ij}, x_{-i}))}{\sum_{k \in \Omega_i} exp(\frac{1}{\epsilon} J_i(e_{ik}, x_{-i}))}$
- Because of this functional form of choice probability function, deterministic noise form of perturbation is preferred for implementing the algorithm.
- The use of perturbed best response in general means that Nash equilibria are no longer the fixed point of the overall stochastic fictitious play algorithm. If $\epsilon > 0$, then the stochastic fictitious play never gives back the NE of the original game.

Algorithm 2 Stochastic fictitious play

a_i, $a_{-i} \leftarrow [\,]$ (# empty list for action of players)
b_i, $b_{-i} \leftarrow [\,]$ (# empty list for belief of players)
$sb_i \leftarrow shape(\Omega_i)$
$sb_{-i} \leftarrow shape(\Omega_i)$
b_i.append($random(sb_i)$) (# initial belief)
b_{-i}.append($random(sb_{-i})$) (# initial belief)
$J_i \leftarrow input(CM)$ (#input for cost matrix)
$J_{-i} \leftarrow input(CM)$ (#input for cost matrix)
$v_i \leftarrow$ random(1) (#initial random entropy)
$v_{-i} \leftarrow$ random(1)
$\epsilon \leftarrow$ random(1)
$i \leftarrow 1$
if $i \leq itr$ **then** (# itr = number of iterations)
    $\tilde{J}_i \leftarrow J_i + \epsilon v_i$ (# perturbed cost)
    $\tilde{J}_{-i} \leftarrow J_{-i} + \epsilon v_{-i}$
    $a_i.append(\text{BR\_per\_s}(b_i, \tilde{J}_i))$ (# append action)
    $a_{-i}.append(\text{BR\_per\_s}(b_{-i}, \tilde{J}_{-i}))$
    $v_i \leftarrow \text{entropy}(a_i[i])$ (# new entropy)
    $v_{-i} \leftarrow \text{entropy}(a_{-i}[i])$
    $z_i \leftarrow \frac{i}{i+1} b_i[i-1] + \frac{1}{i+1} a_{-i}[i]$
    $z_{-i} \leftarrow \frac{i}{i+1} b_{-i}[i-1] + \frac{1}{i+1} a_i[i]$
    $b_i.append(z_i))$ (# append belief)
    $b_{-i}.append(z_{-i}))$
    $i \leftarrow i + 1$.
close
(# BR_per_s gives mixed-strategy, $\widetilde{\Phi}_i$ function)

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.

The game has a pure strategy Nash equilibrium which is both the players confessing. To begin with I would like to show that the algorithm performs for this basic game.

Simulation

Prisoner's Dilemma Game

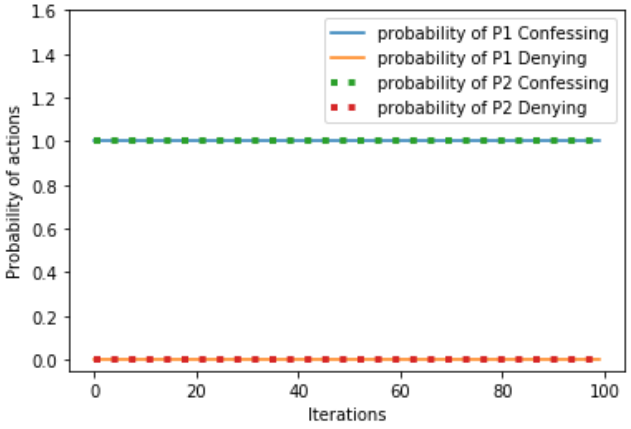


Figure: Actions-FP.

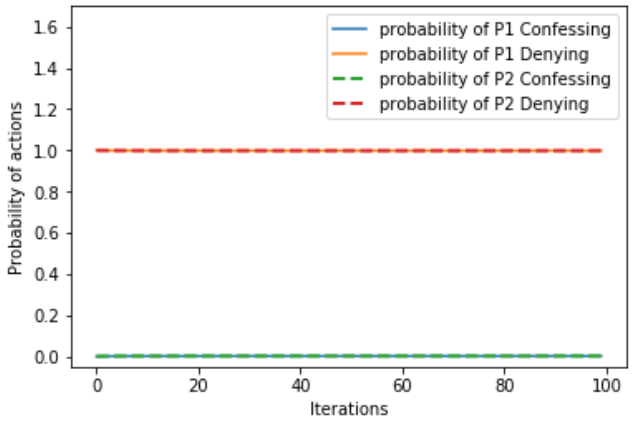


Figure: Actions-sFP.

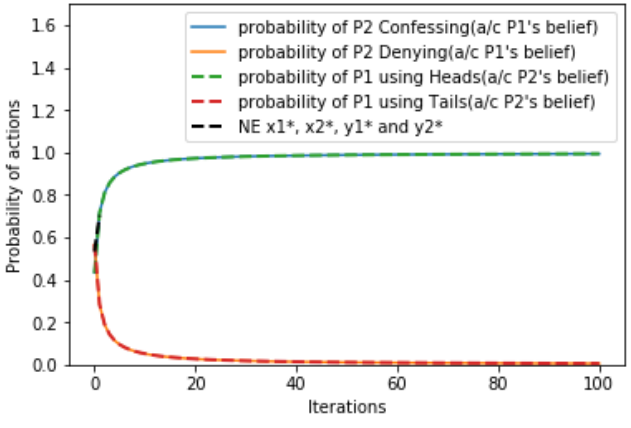


Figure: Beliefs-FP.

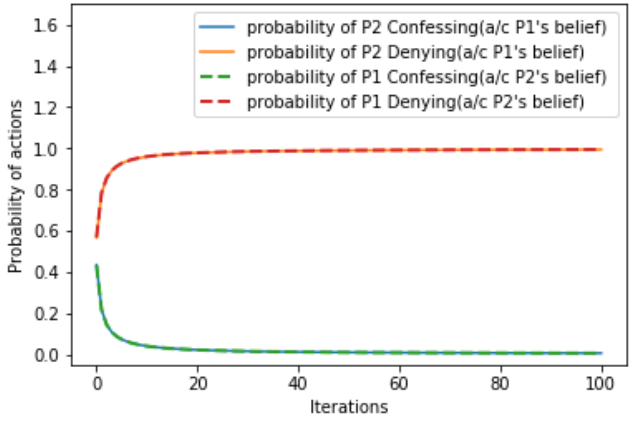


Figure: Beliefs-sFP.

Simulation

Prisoner's Dilemma Game

- Actions and beliefs of both the players in both the algorithms converge for the the Prisoner's Dilemma game.
- FP algorithm converges to NE of the game whereas sFP converges to an approximation of NE of the game.
- The convergence rate for both the algorithm is almost the same and we have a dominant action which is evident by the action profile of the players.

G_2 : Matching Pennies Game

P1 \ P2	Heads	Tails
Heads	-1,1	1,-1
Tails	1,-1	-1,1

Cost Matrix for G_2.

The game has no pure strategy Nash equilibrium and the mixed strategy NE of the game is both the players uniformly selecting one of the two actions. $(x^*, y^*) = ([0.5, 0.5],[0.5, 0.5])$

Simulation

Matching Pennies Game

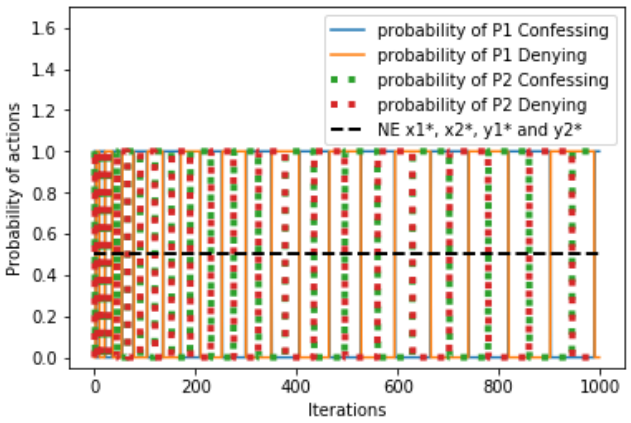


Figure: Actions-FP.

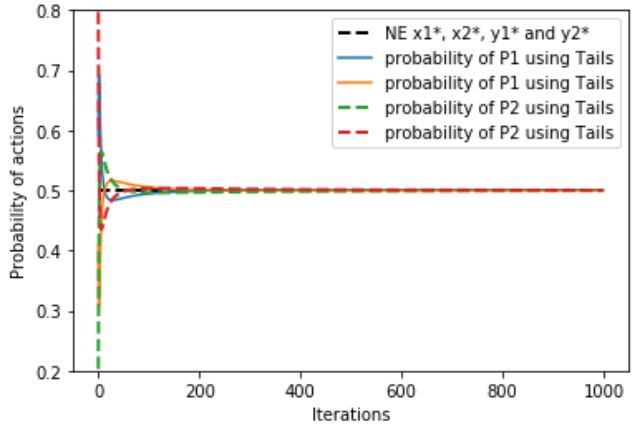


Figure: Actions-sFP.

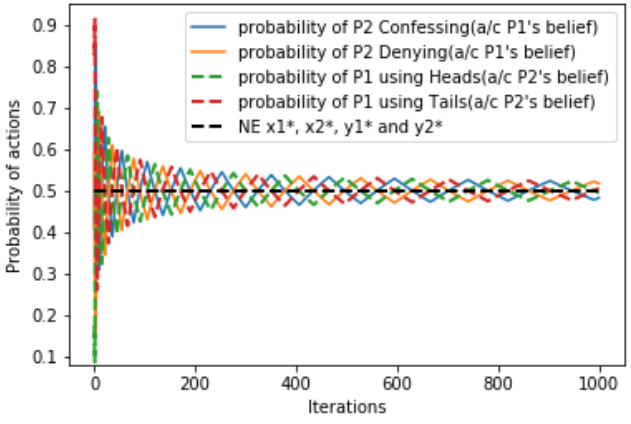


Figure: Beliefs-FP.

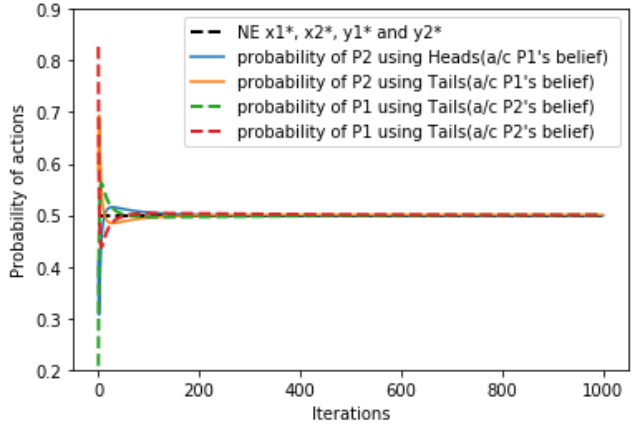


Figure: Beliefs-sFP.

Simulation

Matching Pennies Game

- Actions of the Players do not converge for FP algorithm which was expected since there exists no pure-strategy Nash equilibria for the game.
- beliefs of both the players in both the algorithms converge to the NE of the for the the Matching Pennies game.
- The convergence rate of beliefs in sFP is much faster than the convergence rate of beliefs in FP algorithm.
- sFP also converges to the NE of the game

Simulation

3X3 zero-sum game

G_4 : 3×3 zero-sum Game

P1 \ P2	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.

This game has no pure strategy NE and has a mixed strategy NE. The simulation was performed for a random ϵ which in this scenario was not very close to zero.

Simulation

3X3 zero-sum game

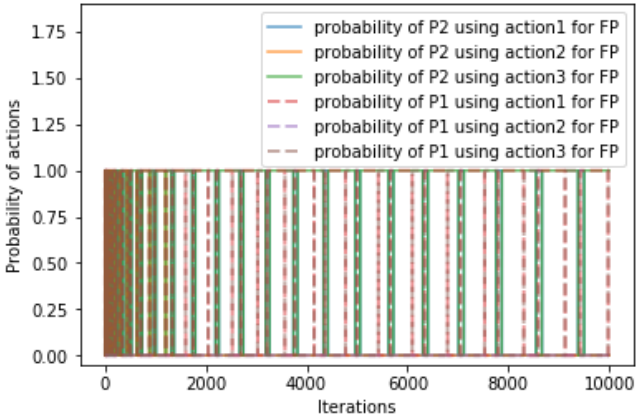


Figure: Actions-FP.

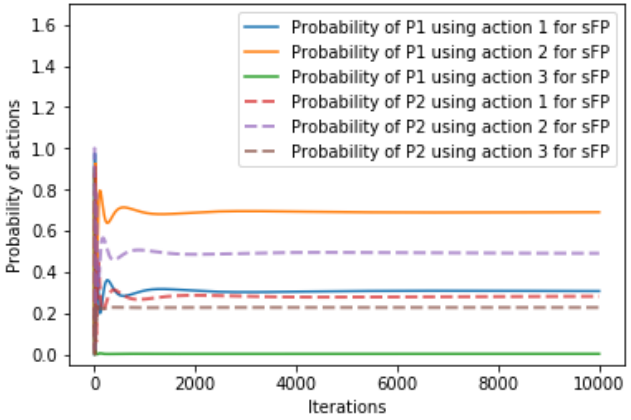


Figure: Actions-sFP.

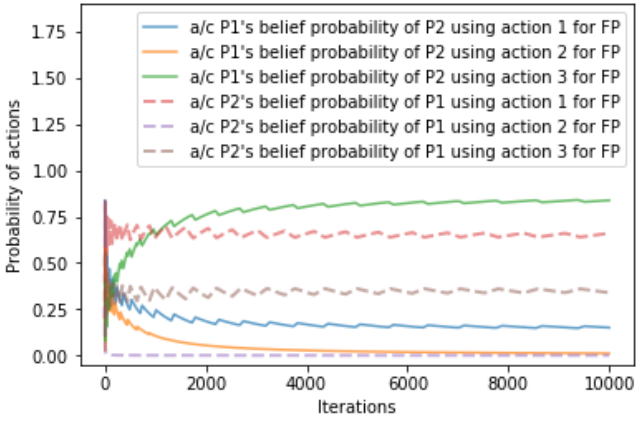


Figure: Beliefs-FP.

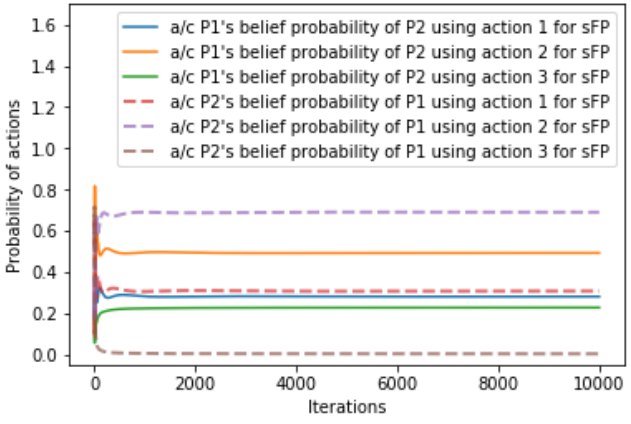


Figure: Beliefs-sFP.

Simulation

3X3 zero-sum game

- The actions of the Players do not converge for FP algorithm.
- Beliefs of the players slowly converges to mixed strategy NE of the game.
- Because of large ϵ there is a considerable amount of deviation in the equilibrium point of the sFP from the NE of the game.
- Convergence rate of the sFP algorithm is much faster than the convergence rate of the FP algorithm.

Comparison

Order Complexity

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 $\mathcal{O}(\text{sFP})$ depends on the $\widetilde{\Phi}_i$ function and the function used to find entropy ($\mathcal{O} \approx s$). $\widetilde{\Phi}_i$ includes, evaluation of expected cost($\mathcal{O} \approx m$) of the players and evaluation of *logit* function($\mathcal{O} \approx m$) .
- $\mathcal{O}(\Phi_i^P) \approx 2 * m$(worst case) since it only involves finding the expected cost of players($\mathcal{O} \approx m$) and then finding the minimum($\mathcal{O} \approx m$) of all the expected costs.
- Overall the complexity of $\mathcal{O}(\text{sFP}) > \mathcal{O}(\text{FP})$ 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. So if we are using different number of iterations for each algorithm then we need multiply the number of iterations to $\mathcal{O}($ one iteration) to get the complexity of the overall algorithm(possibly, (sFP) $< \mathcal{O}(\text{FP})$).

Comparison

Comparison measure \ Algorithms	s-FP	FP
$\mathcal{O}$(iteration)	s+2m	2m
Run-time	T	$t < T$
Fixed point	approximate NE	actual 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 in sFP

Convergence rate of sFP

ϵ ($\epsilon > 0$), in the deterministic perturbation form defines the noise level. As the noise approaches zero in the *logit* choice approaches unperturbed maximization and as ϵ tends to infinity it approaches uniform randomization.

To study the behaviour of convergence rate with respect to ϵ, considering the P2's probability using heads in Matching Pennies game. Using the same initial condition for all the ϵs.

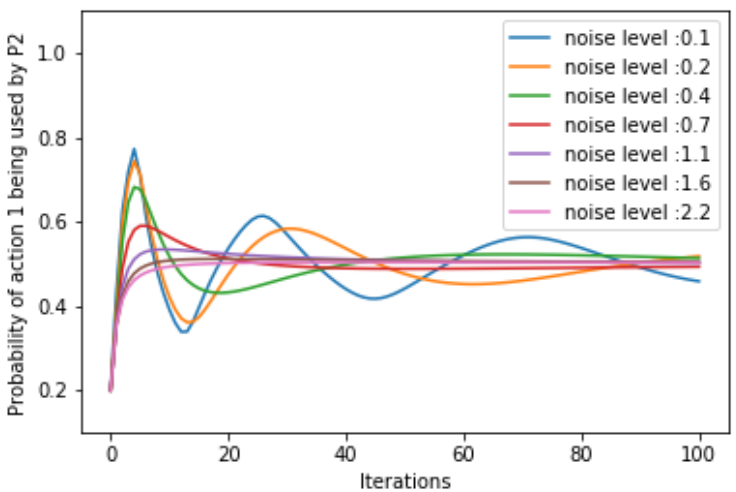


Figure: Beliefs-FP.

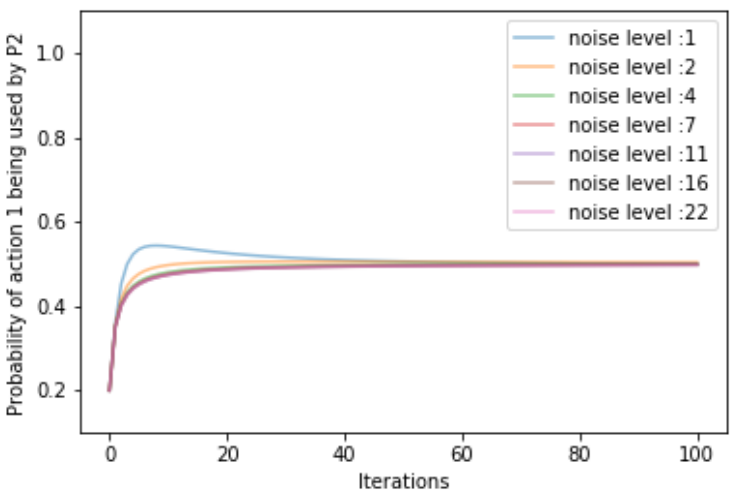


Figure: Beliefs-sFP.

Effect of noise in sFP

Convergence rate of sFP

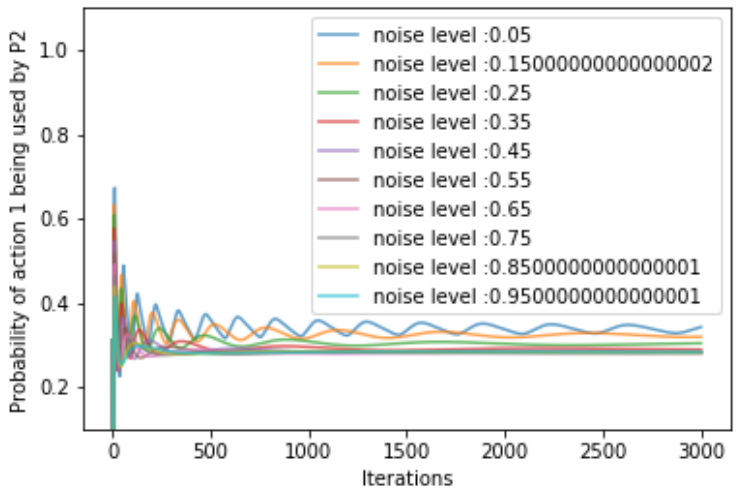
- It can be seen that as the ϵ increases the rate of convergence of the belief in sFP algorithm increases.
- After a sufficiently large ϵ the increase in rate of convergence of the belief is almost insignificant.
- As $\epsilon \to \infty$ the *logit* function tends towards a uniform random distribution and since the uniform random distribution is the NE of the game we see no divergence of the sFP equilibrium point from the NE point.

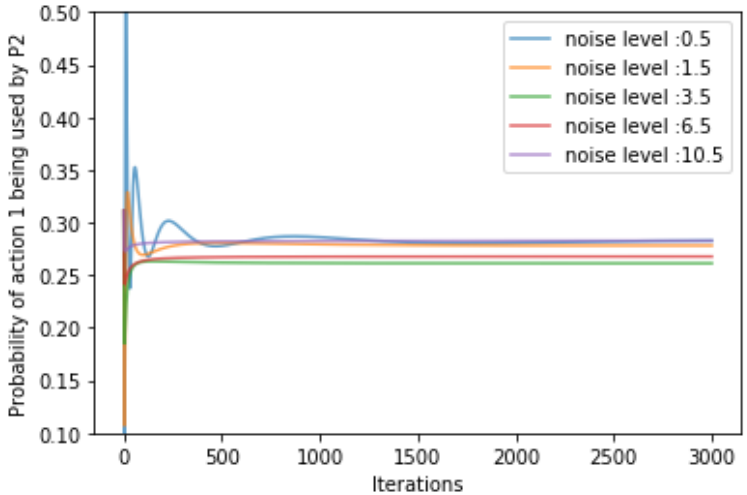
Effect of noise in sFP

Divergence of fixed point of sFP from NE of the Game

ϵ ($\epsilon > 0$), in the deterministic perturbation form defines the noise level. As the noise increases the cost function becomes more and more perturbed and *logit* choice approaches approaches uniform randomization. Thus the beliefs of players deviate from NE to uniform random distribution as $\epsilon \to \infty$.

To study the behaviour of convergence rate with respect to ϵ, considering the P2's probability using action 1 in 3×3 game. Using the same initial condition for all the ϵs.





Effect of noise in sFP

Divergence of fixed point of sFP from NE of the Game

- It can be seen from the plots above that the equilibrium probability of sFP moves farther from the NE of the game as ϵ increases.
- The equilibrium probability of sFP does not increase above $\frac{1}{3}$ which is the probability of P2(when uniformly randomized) using any strategy when $\epsilon \to \infty$

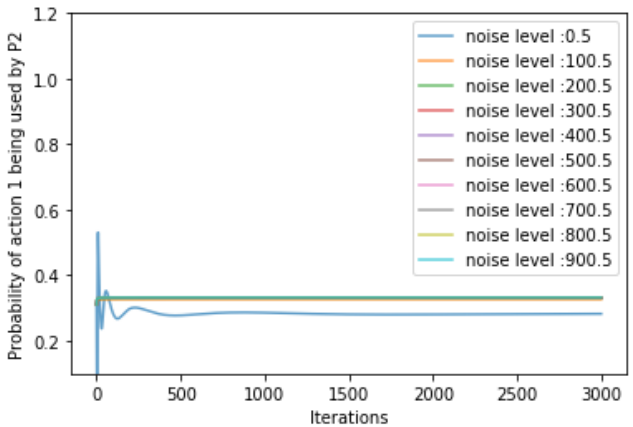


Figure: Beliefs-sFP

- Thus there is a tradeoff between convergence rate and

Summary

- The choice of the noise level ϵ must be made wisely. A significantly large ϵ will in general result to a fixed point far away from the actual Nash equilibria of the game.
- Convergence rate is not the only parameter for the goodness of an algorithm. There are several other aspects like the complexity of the algorithm, run-time of the algorithm and the memory required for implementation which should be considered while comparing the algorithms.
- Each of the two algorithms outperforms the other algorithm in certain measures of the goodness of an algorithm. Thus which algorithm to prefer depends on the situation and the desire of the person trying to implement the algorithm.

Thank You

Questions ?