Game theory –Prisoner's dilemma

Divya Krishnan

Summer 2017

1. Introduction:

The prisoner's dilemma is one of the most popular is one of the most popular games discussed and analyzed in game theory.

The dilemma was first discussed by mathematicians Merrill Flood and Melvin Dresher in 1950 .Prisoner's dilemma helps us in understand as to why co-operation is necessary in society and how blind pursuit towards self interest by all the agents in the society can have a disastrous effect. It basically helps us in understanding the balance of co-operation and fierce competition in politics, society and biology.

2. Prisoner's dilemma

The basic prisoner's dilemma revolves around 2 prisoners who have been caught by the police however the police don't know who the real culprit is .Both the prisoners are interrogated separately. They have an option of either confessing for the crime or remaining silent .Even though it appears that the game is independent, it is not and the final verdict would depend on the action of both the prisoners.

If prisoner 1 confesses and prisoner 2 confesses too they get a punishment of 2 years. If either of them confesses or one betrays then the confessor get an imprisonment for 5 years whereas the other is free and if both end up betraying then they get an imprisonment of 3 years

	Prisoner 1 co operates	Prisoner 1 betrays
Prisoner 2 co-operates	2 years	Prisoner 1 =free Prisoner 2=5 years
Prisoner 2 betrays	Prisoner 1 =5 years Prisoner 2=free	3 years

In pursuit of self interest both the players betray each other and get a higher term of prison compared to 2 years that they would have got had they co-operated.

The simplest form of PD can be explained by the payoff matrix in the form of -

	C(confess)	D(defect)
C(confess)	R,R	S,T
D(defect)	T,S	P,P

Satisfying the condition: $T>R>P>S$

We see from the table that mutual defection is the only Nash equilibrium in the game. The dilemma in the game is that though mutual co-operation has higher yield than mutual defection it's not rational outcome due to personal self interest of the player

2.1: Iterated Prisoner's Dilemma:

If two players play PD more than once in succession and they remember previous actions of their opponent and change their strategy accordingly, the game becomes an Iterative Prisoner's Dilemma An IPD can be finite or infinite, definite or indefinite, and even noisy .As there is no way to understand your opponent's approach in a one-time interaction, the much more interesting problem in an iterated Prisoner's dilemma

The payoff in a PD can be calculated by the following

Payoff(Player1) = ((R+P-S-T)/4)*(1+s2)*(1+s1) + ((S-P)/2)*(1+s1) + ((T-P)/2)*(1+s2) + P

Payoff(Player2) = ((R+P-S-T)/4)*(1+s1)*(1+s2) + ((S-P)/2)*(1+s2) + ((T-P)/2)*(1+s1) + P

Here s1 and s2 denote action of player 1 and player 2 and they can take value 1(confess) and -1(defect)
For further discussion the values R=1,T=1.5,S=0,P=0 is fixed .
of code

Prob1(t) = ((p1+p4-p2-p3)/4)*(1+s1(t-1))*(1+s2(t-1)) + ((p2-p4)/2)*(1+s1(t-1))+((p3-p4)/2)*(1+s2(t-1)) + p4;
Prob2(t) = ((q1+q4-q2-q3)/4)*(1+s1(t-1))*(1+s2(t-1)) + ((q2-q4)/2)*(1+s2(t-1))+((q3-q4)/2)*(1+s1(t-1)) + q4;

s1(t)=sign(Prob1(t)-rand);
s2(t)=sign(Prob2(t)-rand);

Here p1, p2, p3,p4 ,q1, q2, q3 and q4 will each take one bit from the binary representation of the 16 strategies in a 2 player PD

Tit for Tat: Tit for tat is one of the most famous strategies in prisoner's dilemma. In this one player copies the action the other player had taken in the previous round. The binary representation of tit for tat is 1010.

The graphs below represent two players playing tit for tat strategy for 20 iteration The strategy requires the first action of both the players to be confession. For a noiseless condition they payoff of the two players will be as shown in graph 1.a And their action would look like fig 1.b

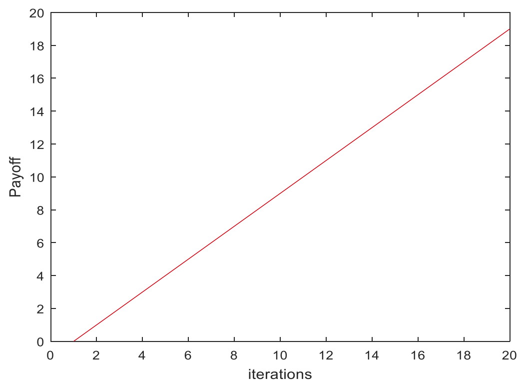


Fig1.a

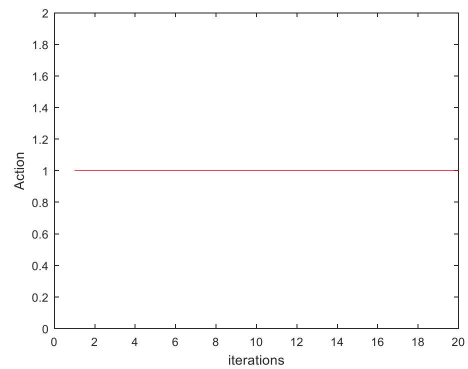


Fig 1.b

Fig1.c and 1.d shows the payoff and the action when a player play a -1 under certain circumstance (noise)

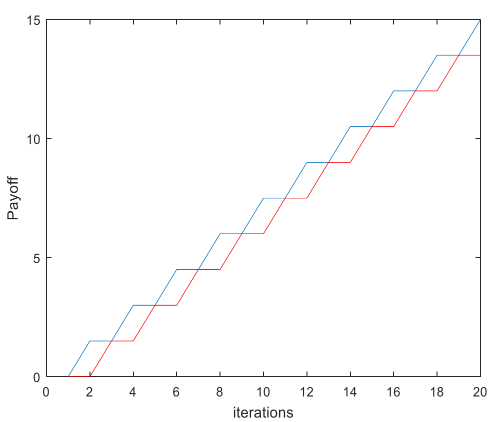


Fig 1.c

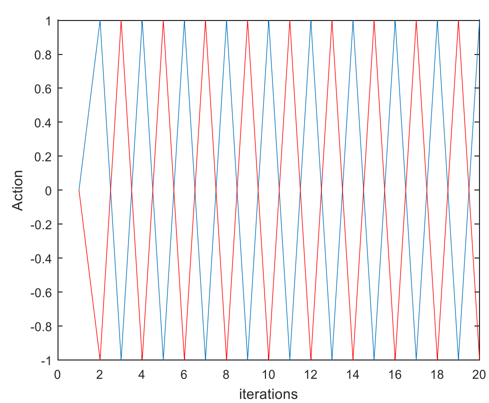


Fig1.d

Pavlov Strategy: Pavlov means that if other player co-operated in the last move then repeat your previous move but if not then switch to the other move. It means when reward or temptation is received keep on repeating last move else switch. Pavlov is represented by 1010 in binary form. Like TFT Pavlov strategies with cooperation and will punish the opponent immediate when the opponent defects .However unlike TFT which slides to continuous defection Pavlov will try to switch to co-operation .So Pavlov basically works like an error correction and if both players play Pavlov they will go for mutual co-operation .
The graphs 2.a and 2.b show Pavlov action and payoff assuming both initial actions were cooperation

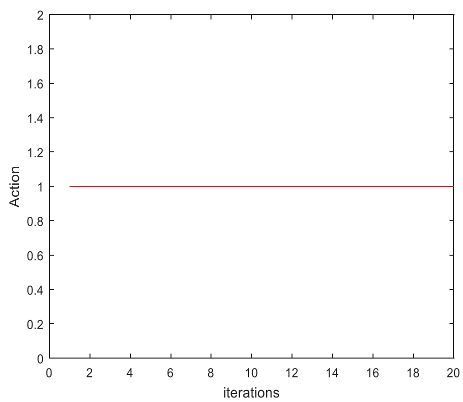


Fig2.a

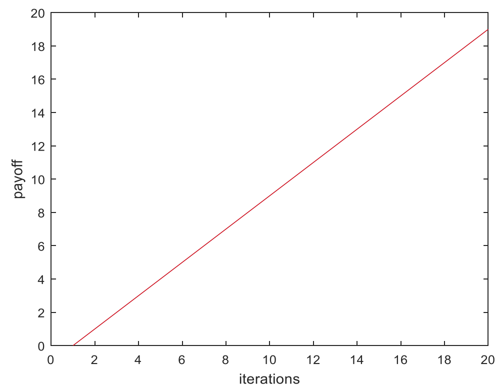


fig2.b

Assuming player 2 accidently defects

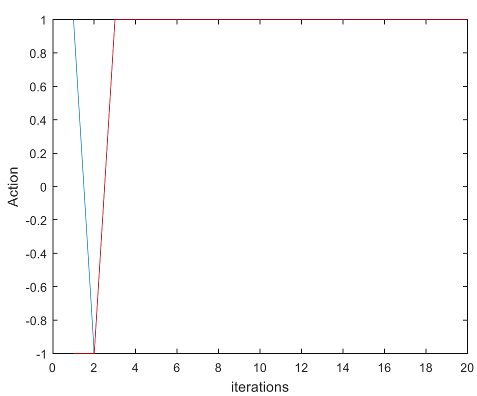


Fig 2.c

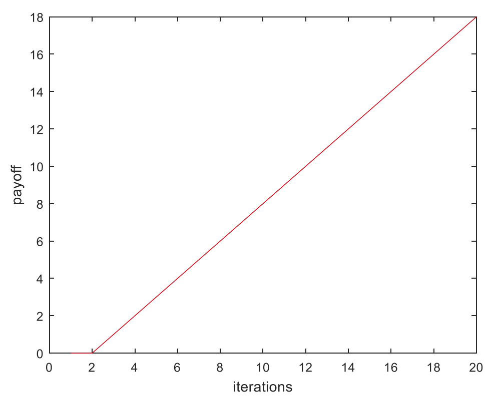


Fig 2.d

2.2Noise and its effects:
Noise can affect the conventional working of a strategy and adds an uncrtainity factor in calculation of payoff and selection of action.
Noise can be classified into two main types:

1) ***<u>Noise in execution :</u>***
Agents could make mistakes in implementing their intended strategies
E.g.: Player 1 must have intended in playing co-operation but must have accidently chosen to defect.
This can be achieved my multiplying a -1 to their intended action
So if a player had actually chosen to cooperate it will end u defecting due to noise factor.

This is done by the following line of code
sa1(t)=s1(t)*sign(1/(1+exp(-1/T1))-rand)
sa2(t)=s2(t)*sign((1/(1+exp(-1/T1)))-rand)
S1 and s2 are noiseless action in the code.If T1 =0 it is a no noise ,ideal situation .However if T1=infinity we have a -1 being with a probabilty of 0.5

Essentially this means that even though the player may be using some strategy, the implementation noise reduces it to a purely random choice.
This means that the Payoff1 may also change, as the s1 used in calculating the Payoff is not the original s1 obtained based on strategy but the one obtained because of the mistaken implementation.

So the actual payoff equation are as follows:
Payoff of agent1 Payoff1(t) = ((R+P-S-T)/4)*(1+sa1(t))*(1+sa2(t)) + ((SP)/2)*(1+sa1(t)) +((T-P)/2)*(1+sa2(t))+P

Payoff of agent1 Payoff2(t) = ((R+P-S-T)/4)*(1+sa1(t))*(1+sa2(t)) + ((SP)/2)*(1+sa2(t)) +((T-P)/2)*(1+sa1(t))+P

<u>2) Noise in Communication:</u> Here, even if player 1 correctly implements his intended action (according to the strategy being used) it may be misinterpreted by the other player.
So, for example, even if player1 plays cooperation, under noise player 2 may think that agent 1 has defected. However, the Payoff of player 1 is unchanged from the original noiseless condition as the agent 1 has actually correctly implemented the action, but player2 will record it as a different action and will use this mistaken record in deciding the action for the next iteration based on his strategy.
This can be done using the code:
sm1=s1*sign(1/(1+exp(-1/T2)))-rand)
sm2=s2*sign(1/(1+exp(-1/T2)))-rand)
Here sm1 and sm2 are action due to communication noise and T2 is parameter that decides noise in communication

When both noises play a role both communication and execution error affect the selection of the next action but only Implantation error will affect the final payoff calculation
So in this situation the probability of selection of next action final payoff calculation is :

$$Prob1(t) = ((p1+p4-p2-p3)/4)*(1+sa1(t-1))*(1+sm2(t-1)) + ((p2-p4)/2)*(1+sa1(t-1)) + ((p3-p4)/2)*(1+sm2(t-1)) + p4;$$

$$Prob2(t) = ((q1+q4-q2-q3)/4)*(1+sm1(t-1))*(1+sa2(t-1)) + ((q2-q4)/2)*(1+sa2(t-1)) + ((q3-q4)/2)*(1+sm1(t-1)) + q4;$$

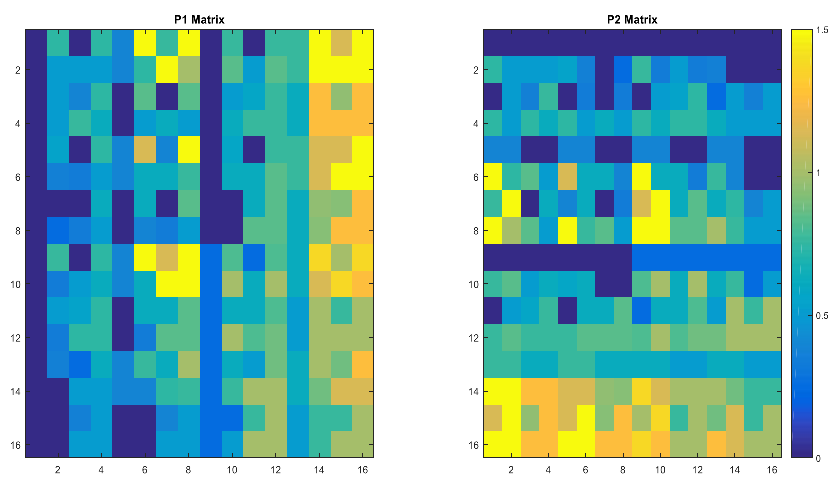
```
sm1=sa1*sign(exp(1/(1+exp(-T2)))-rand);
sm2=sa2*sign(exp(1/(1+exp(-T2)))-rand);
```

Payoff of agent1 $Payoff1(t) = ((R+P-S-T)/4)*(1+sa1(t))*(1+sa2(t)) + ((SP)/2)*(1+sa1(t)) +((T-P)/2)*(1+sa2(t))+P$

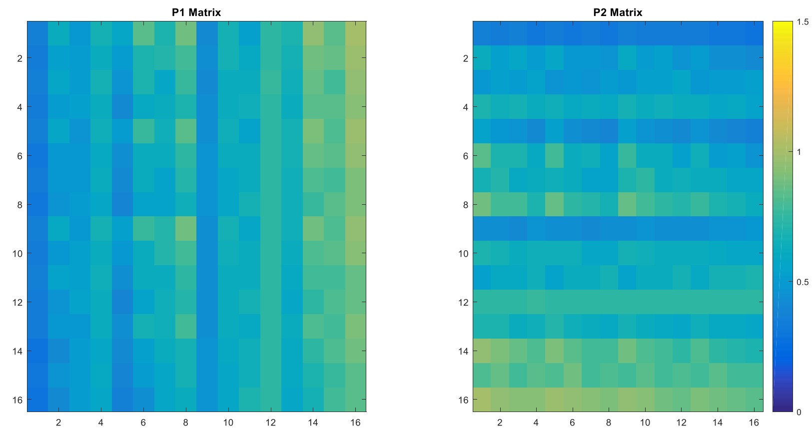
Payoff of agent1 $Payoff2(t) = ((R+P-S-T)/4)*(1+sa1(t))*(1+sa2(t)) + ((SP)/2)*(1+sa2(t)) +((T-P)/2)*(1+sa1(t))+P$

16X16 strategies and 2 Player PD :

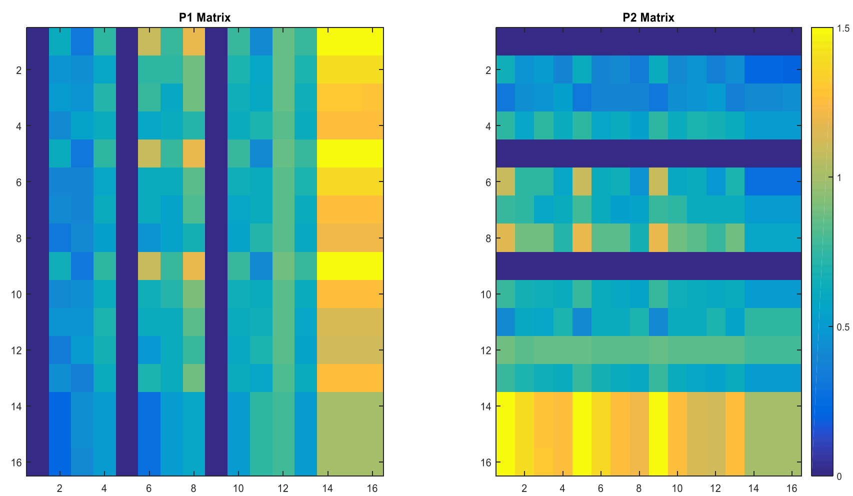
To analyze performance of all 16 stratégies against each other for both Player 1 and Player2. Below are the images of the matrices of player1 and player2 playing all 16 strategies
We notice that, the two matrices are transpose of each other which was desired
The left side denotes player 1 playing 0 to 15 stretegies (its named from 1 to 16 so subtract 1 from the column number to get the real strategy number) and the below repesents player 2 playing stratégies



T1=1 and T2=0

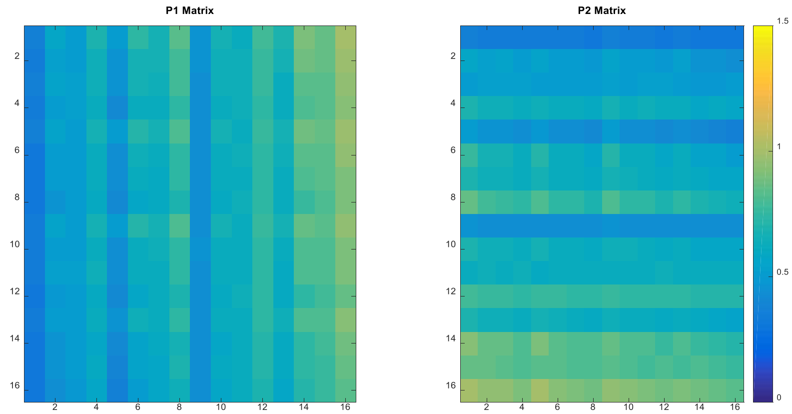


T1=0 and T2=1



Notice that strategy 1000 and strategy 0100 (corresponding to number 9 and 5) are do well when played by a player, as it results in the other player getting a bad payoff.

At T1=1 and T2=1



How many co-operate ?

Now we consider player 1 surrounded by k other players
Player 1 will choose his strategy based on how many players co operated with it.
So it will have 2^2(k+1) strategies to select from. If we consider k=4 player 1 will have 1024 strategies. Also we assume here that all other players here play the same strategy and will play a 2 player PD with player 1. They will have a bag of 16 strategies to select from. Hence we get a matrix of 1024x16.