Axelord

February 21, 2019

Author: Guangya Wan Time : Feb 21 2019

1 Introduction

1.0.1 Axelord's tournament is hold back to 1970s; it's held by professor Axelord who wish to find the best way to deal with iterated prisioner's dilemma. Among more than 60 participants, it turns out that the Tit-for-tat designed by Anatol Rapoport is the winner, and it is also the simplest one

1.0.2 Tit-for-tat have lots of variations(though they did not win the original version in the tournaments), I will show the effects of the original tit-for-tat and one of its variations -- Grudger here.

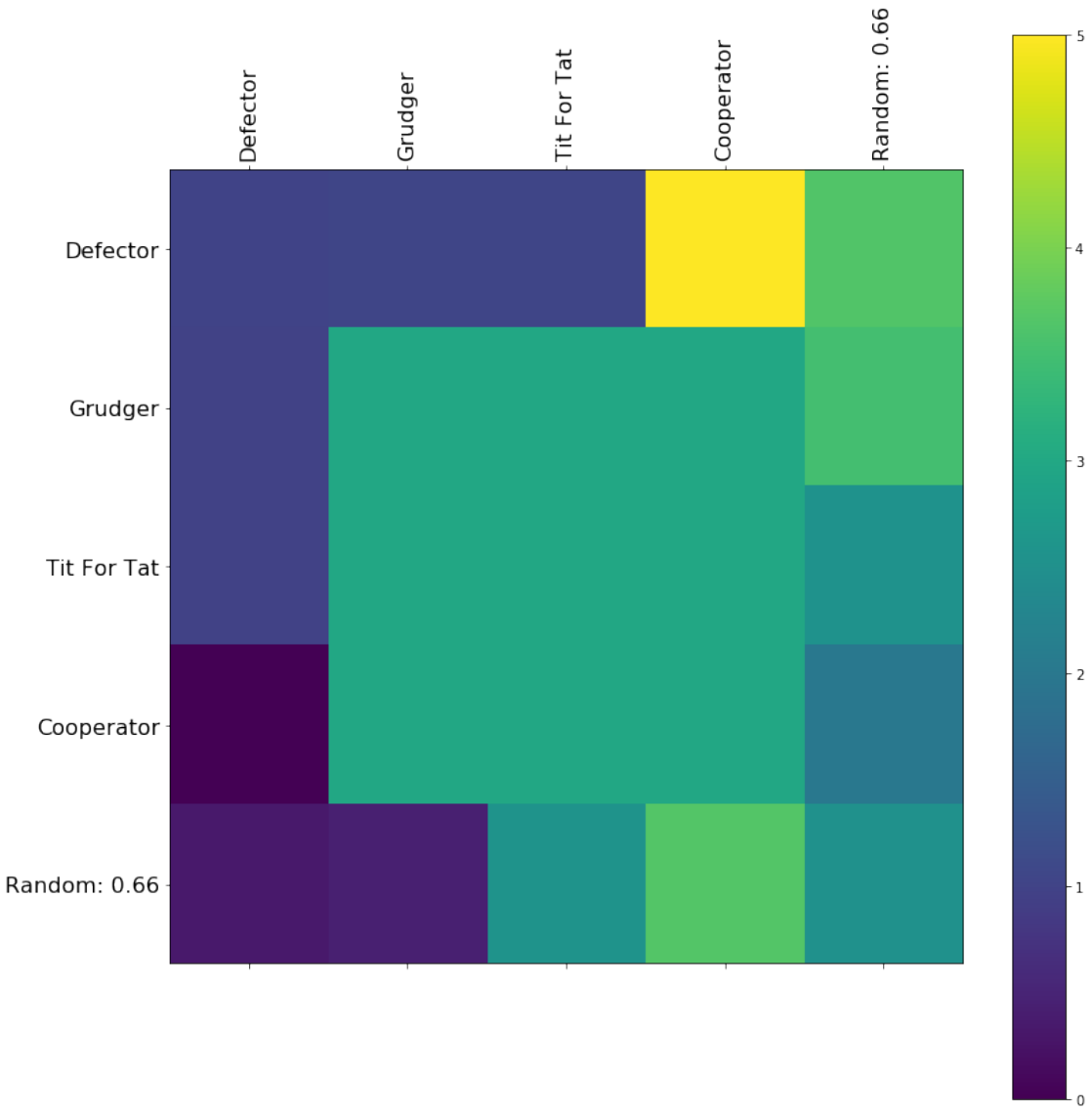
1.0.3 Tit for tat is to cooperate first, and do whatever its opponents do last time. Grudger is basically the same, except that it does not forgive. So as long as the opponent defects, it will always defect in the future rounds.

1.0.4 The top ranked stretegies in Axelord's tournaments have genereally 4 properties : Nice, Retaliatation, Forgiveness, and non-envious(Not trying to get higher than your opponents) However, this really depends on the enviroments of the tournaments, one can not simply give tit-for-tat and expected to win every single time. I will show some examples below

2 Simulating matches

```
In [1]: import axelrod as axl
        import pprint

In [2]: players = [axl.Cooperator(), axl.Defector(),
                   axl.TitForTat(), axl.Grudger(),axl.Random(0.66)]
        # grudge is similar to titfor tat expect it does not forgive
        tournament = axl.Tournament(players,turns = 100, repetitions = 10)
        results = tournament.play()
        plot = axl.Plot(results)
        p = plot.payoff()
        p.show()
        summary = results.summarise()
        pprint.pprint(summary)
```

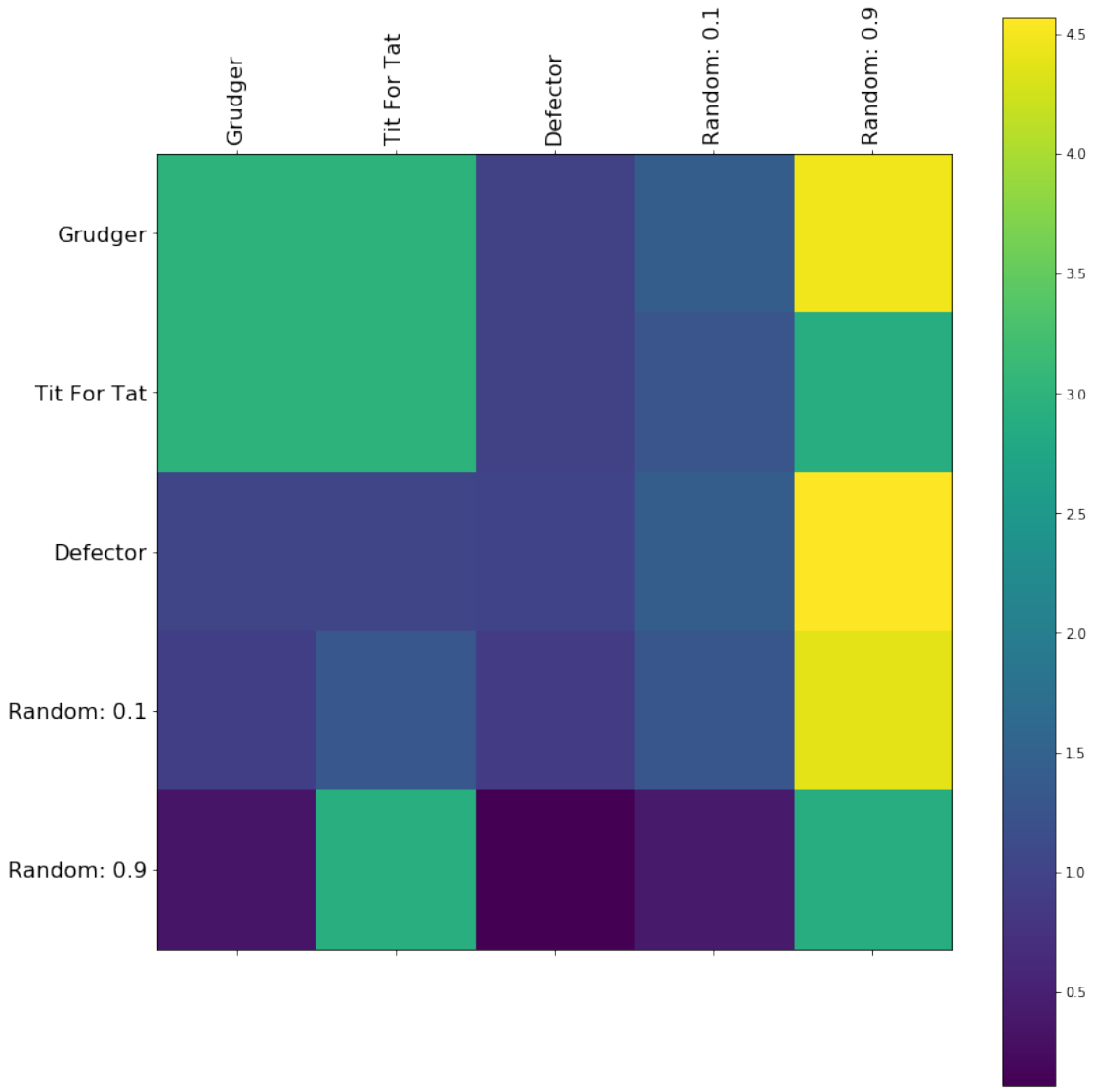
```
Playing matches: 100%|| 15/15 [00:00<00:00, 52.53it/s]
Analysing: 100%|| 25/25 [00:00<00:00, 124.88it/s]
/home/wan/anaconda3/lib/python3.7/site-packages/matplotlib/figure.py:457: UserWarning: matplotl
  "matplotlib is currently using a non-GUI backend, "


[Player(Rank=0, Name='Defector', Median_score=2.6950000000000003, Cooperation_rating=0.0, Wins=
 Player(Rank=1, Name='Grudger', Median_score=2.62, Cooperation_rating=0.50825, Wins=1.0, Initia
 Player(Rank=2, Name='Tit For Tat', Median_score=2.385, Cooperation_rating=0.6705, Wins=0.0, In
 Player(Rank=3, Name='Cooperator', Median_score=2.0025, Cooperation_rating=1.0, Wins=0.0, Initi
 Player(Rank=4, Name='Random: 0.66', Median_score=1.7575000000000003, Cooperation_rating=0.657
```



In [4]:
```
players = [axl.Defector(),axl.TitForTat(),
           axl.Grudger(),axl.Random(0.1),axl.Random(0.9)]
# grudge is similar to titfor tat expect it does not forgive
tournament = axl.Tournament(players,turns = 100, repetitions = 10)
results = tournament.play()
summary = results.summarise()
plot = axl.Plot(results)
p = plot.payoff()
p.show()
summary = results.summarise()
pprint.pprint(summary)
```

```
Playing matches: 100%|| 15/15 [00:00<00:00, 29.95it/s]
Analysing: 100%|| 25/25 [00:00<00:00, 177.75it/s]
```

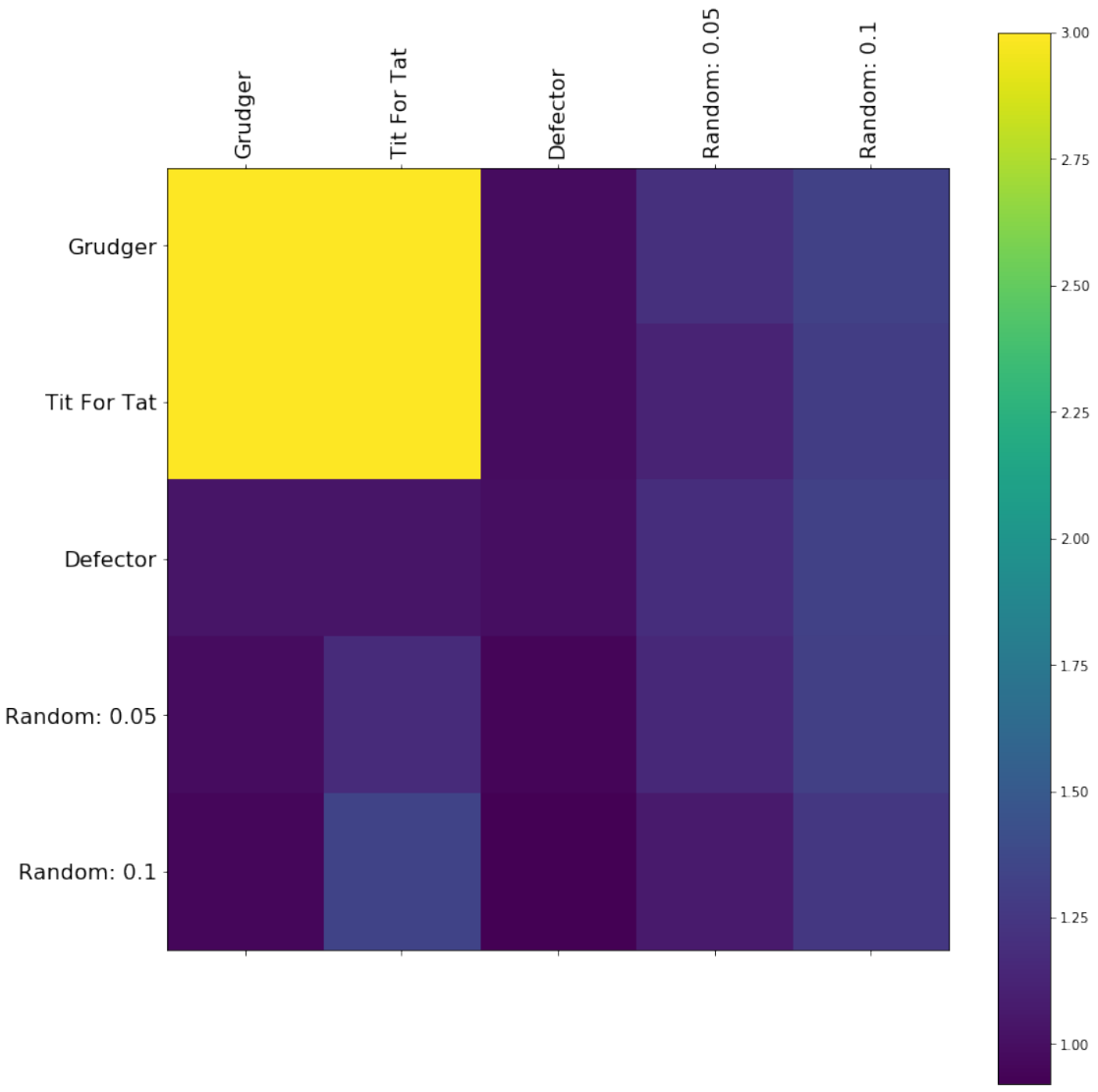
```
[Player(Rank=0, Name='Grudger', Median_score=2.4675000000000002, Cooperation_rating=0.27725, W
 Player(Rank=1, Name='Tit For Tat', Median_score=2.04375, Cooperation_rating=0.5065, Wins=0.0,
 Player(Rank=2, Name='Defector', Median_score=2.015, Cooperation_rating=0.0, Wins=4.0, Initial
 Player(Rank=3, Name='Random: 0.1', Median_score=1.8875, Cooperation_rating=0.10325, Wins=2.0,
 Player(Rank=4, Name='Random: 0.9', Median_score=0.9299999999999999, Cooperation_rating=0.9007
```




```
In [5]: players = [axl.Defector(),axl.TitForTat(), axl.Grudger()
                  ,axl.Random(0.1),axl.Random(0.05)]
        # grudge is similar to titfor tat expect it does not forgive
        tournament = axl.Tournament(players,turns = 100, repetitions = 10)
        results = tournament.play()
        summary = results.summarise()
        plot = axl.Plot(results)
        p = plot.payoff()
        p.show()
        summary = results.summarise()
        pprint.pprint(summary)

Playing matches: 100%|| 15/15 [00:00<00:00, 37.91it/s]
```

```
Analysing: 100%|| 25/25 [00:00<00:00, 141.11it/s]
```

```
[Player(Rank=0, Name='Grudger', Median_score=1.6375, Cooperation_rating=0.25775, Wins=2.0, Ini
 Player(Rank=1, Name='Tit For Tat', Median_score=1.6087500000000001, Cooperation_rating=0.2947
 Player(Rank=2, Name='Defector', Median_score=1.145, Cooperation_rating=0.0, Wins=4.0, Initial
 Player(Rank=3, Name='Random: 0.05', Median_score=1.1075, Cooperation_rating=0.0475, Wins=2.0,
 Player(Rank=4, Name='Random: 0.1', Median_score=1.0775000000000001, Cooperation_rating=0.0895
```



3 Conclusion

3.0.1 There are no stretegies either mathmatical or emprically proved to win all the iterated games right now. It dependes on the ecology of the enviroments.In a tournaments where most of players perfer to be "nice", a defector can usually be ranked as 1st. In a game where most players like to defect, tit-for-tat or its variation are often the winner

3.0.2 There are other better stretegies nowdays where in a game you can assign mutiple players. You can assign a predetermined winner and sacrifice the rest of other players. Specifically, they are programmed to able to recoginize the winner. If it's winner, they will always cooperate, otherwise they will defect to lower their opponents' scores

4 Discussion

4.1 I think the probability of decision of opponents(if they are random) can be described by binomal distirbution or bayes network(if they are dependent on last move), so very useful information(like we see in the summary above) can be retrieved after many rounds. It will be often meaningless at the beginning of the game since the sample is too small, however, after a reasonable amounts of games, Statistics can be powerful.

4.2 So My stretegy is to play random for the first 30 rounds in order to find opponents' distribution and then using some probability and statistics techniques to maximize our scores to outperform tit-for-tat, what are your opinions?

In [ ]:

In [ ]: