The Prisoner’s Dilemma

why it rewards to be nice

Alberto Nicoletti

June 2024

Abstract

In this project I developed a program to make a tournament in which different strategies compete in the famous Prisoner's Dilemma. This work takes inspiration from the 1984 study By Robert Alexrod[1], the game *Evolution of Trust* by Nick Case[2] and the Veritasium's video, *"What Game Theory Reveals About Life, The Universe, and Everything"*[3].

Contents

1 Introduction: The Problem

The Prisoner's Dilemma is a game theory experiment, originally invented in 1950. One common description of the problem involves two criminals of the same gang that are arrested and placed in two different cells with no way of communicating with each other. Each of them has then to choose to either testify against his colleague or stay silent. Each option gives different benefit or penalties depending on the choices of both prisoners.
As a reference, I will use the following, more general, description:

> You are playing a game in which the goal is to make the highest amount of points. You have two possible actions: **cooperate** or **defect**.
> This is a two players game, player A and player B. You are player A and you have no way of communication with player B. If your choice is to cooperate and also player B cooperates you both get 3 points, but if he defect you get 0 point and player B gets 5 points. If instead you decide to defect and player B's choice is to cooperate, you get 5 points and player B gets 0 points, but if you both defect you get 1 point each.
> Table 1.1 shows the scores for the possible outcomes of the game.

A \ B	Cooperate	Defect
Cooperate	B:3 A:3	B:5 A:0
Defect	B:0 A:5	B:1 A:1

Table 1.1: The benefits and penalties scores of the Prisoner's Dilemma.

It seems that the best choice is to defect, as it will guarantee you to have one point and hopefully you will even get five. But there is one small change that will make every different: repetition. If instead of playing this game once, we repeat it in multiple rounds, adding the overall points, there will be different possible strategies to get the highest possible number of points and things become more interesting.

What makes this game interesting is that it generalizes and simplifies many real-world situations, from international politics to everyday human behaviours. I believe that the Prisoner's Dilemma is a mathematical model that can be used to answer the question: *"should we be nice with each other?"*.

2 The Tournament

In 1984, Robert Axelrod, Professor of Political Science and Public Policy at the University of Michigan, invited experts in game theory to submit programs able to play the Prisoner's Dilemma and he made these programs compete against each other on a computer tournament of the game. The results showed that the winning program was quite simple, implementing a strategy that we can call *Tit for Tat*.
In this project I developed a program in *c++* that implements a similar tournament. My goal is to test and play with some possible strategies and let the code available to whoever will like to try to experiment with this problem. I implemented 16 strategies and a main program that makes the tournament. In the tournament, each strategy plays once against each other strategy, including a copy of itself. Each game consists in 100 moves. Summing all the points at the end of the tournament we can see which are the best strategies.

3 Strategies

In this section I will list the strategies implemented with a short description.

- *Always Nice:* Always cooperates.
- *Always Mean:* Always defects.
- *Tit for Tat:* The first move is cooperate, then in each move copies the last move of the opponent.
- *Nice not Twice:* The first move is cooperate, then it keeps cooperating until the opponent defects for a second time (even not consecutive). In this case makes one defecting move and repeat the process.
- *Random:* Each move is selected at random with a coin flip.
- *You Didn't:* Start with cooperation and keeps cooperating until the opponent makes one defection move. From this point on always defects.
- *I Remember:* The first move is defect. Then, every move is the most common move among the opponent's choices.
- *Increasing:* Starting with cooperation, it alternates a cooperation phase and a defection phase. The cooperation phase consists always in one move, while the defection phase increase by one move each time. For example, c-d-c-d-d-c-d-d-d-c-d-d-d-d-c....
- *Tit for 2 Tat:* Always cooperate but if the opponent defects two consecutive times, it does one defection move.
- *Detective:* The first four moves are always: cooperate, defect, cooperate, cooperate. If during these four move the opponent always cooperated, acts as *Always Mean*. Otherwise, acts as *Tit for Tat*.
- *Simpleton:* The first move is cooperate. Then, if the opponent cooperates, *Simpleton* repeats its own last move. If the opponent defects, *Simpleton* does the opposite of its own last move.
- *Super Detective:* It acts like the detective, but it keep checking the last four opponent moves to decide how to act next.
- *Mean Schizophrenic:* Every three moves it changes strategy, alternating between: *Always Mean*, *You Didn't* and *Increasing*.

- *Nice Schizophrenic:* Every three moves it changes strategy, alternating between: *Always Nice*, *Tit for 2 Tat* and *Nice not Twice.*
- *Twin for Tat:* The first move is cooperate. Then, it does the opposite of the opponent's last move.
- *Not Forever:* Plays at *Tit for Tat.* If the opponent defected 20 times, switch strategy to *Always Mean.*

4 Implementation

This section is dedicated to some technical details about the implementation. The code is available in the repository[1].

4.1 The Tournament

The tournament consists in a double nested *for* loop that makes each strategy play against each other strategy (and also against itself). During each game every move is printed on the *games.txt* file and the final scored is stored in a two dimension array *scores* used to store how many points each strategy made playing against each other strategy. The score of each game is also printed on the file *games.txt*.

At the end of the tournament the program computes the average scored made by each strategy and sort all strategies by this value in descending order. The final result is then printed on the file *results.txt*.

4.2 The Strategies

Each strategy is implemented as a function. The function is called every time one player has to make an action, giving some information as input and returning the chosen move (cooperate or defect) as output. The input (parameters) of the function is:

- prev_moves: an two-dimensional array storing the previous moves of both players.
- n_prev_moves: how many moves have already been done.
- tot_moves: how many total moves are played in the tournament[2].
- current_player: used to indicate which row of the array prev_moves refers to the strategy itself.

We can see that the only information available to each strategy is the list of previous moves of the opponent and of itself. Based on this information the strategy will select its next move and give it as output of the function.

[1]https://github.com/nicoalbe/cooperation

[2]Note: this parameter is only used to handle with the dimension of the array of the moves. In this environment we assume that each player does not know how long each game will be, it has to act as if it may go on forever.

5 Results

Different runs of the programs gave slightly different results, due to the unpredictability of the *Random* strategy. Still, over multiple runs of the experiment, the first five position never changed and the other position suffer minor changes. We use the following result as a sample to make some comments and try to draw some conclusions.

#1: *Not Forever* with 244.875 points
#2: *You Didn't* with 242 points
#3: *Tit for Tat* with 230 points
#4: *Tit for 2 Tat* with 225 points
#5: *Detective* with 221 points
#6: *Nice Schizophrenic* with 214 points
#7: *Nice not Twice* with 214 points
#8: *Simpleton* with 212 points
#9: *I Remember* with 208 points
#10: *Always Mean* with 207 points
#11: *Increasing* with 205 points
#12: *Random* with 202 points
#13: *Mean Schizophrenic* with 201 points
#14: *Twin for Tat* with 197 points
#15: *Super Detective* with 191 points
#16: *Always Nice* with 187 points

6 Conclusion

The experiment confirmed Axelrod's results: the winner of the tournament is a strategy that is a modified version of *Tit for Tat*.

We are are used to games where two or more players compete against each other and one can only win by making more points than the opponents. But life is rarely like this, we often find ourselves in situations in which there is not one winner, this is way this game is particularly interesting.
The Prisoner's Dilemma can show us, with empirical results, that cooperating and being nice to each other is the best strategy in our life. As we learn to view each other as common players of the same game and not as opponents, we can achieve better results and improve our quality of life. What we can learn for the *Tit for Tat* strategy is that we should first of all have trust in each other and keep cooperating as long as others are also cooperating. This tournament's winning strategy is *Not Forever*, and this teaches us that if someone is often mean to us, we should be forgiving for a little bit but then we should just stop being nice with them.

In this report I tried to explain just some of my thoughts, but there is so much more to talk about! I highly recommend the reader to check the bibliography.

Bibliography

[1] R. Axelrod, "The evolution of cooperation*," 1984.

[2] N. Case, "The evolution of trust."

[3] Veritasium, "What game theory reveals about life, the universe, and everything."