# Understanding responses to environments for the Prisoner's Dilemma: A machine learning approach

Nikoleta E. Glynatsi

Month 2020

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy.

School of Mathematics
Ysgol Mathemateg

# Contents

# List of Figures

# List of Tables

# Chapter 1

# A meta analysis of tournaments and an evaluation of performance in the Iterated Prisoner's Dilemma.

The research reported in this Chapter has lead in a manuscript, entitled:

**"Properties of winning Iterated Prisoner's Dilemma strategies"**

Available at: `https://arxiv.org/abs/2001.05911`

Associated data set: [?]

Axerod-Python library version: 3.0.0

The manuscript's abstract is the following:

Researchers have explored the performance of Iterated Prisoner's Dilemma strategies for decades: from the celebrated performance of Tit for Tat, to the introduction of the zero-determinant strategies, to the use of sophisticated learning structures such as neural networks, many new strategies have been introduced and tested in a variety of tournaments and population dynamics. Typical results in the literature, however, rely on performance against a small number of somewhat arbitrarily selected strategies in a very small number of tournaments, casting doubt on the generalisability of conclusions. We analyze a large collection of 195 typically known strategies in 45686 tournaments, present the top performing strategies across multiple tournament types, and distill their salient features. The results show that there is not yet a single strategy that performs well in diverse Iterated Prisoner's Dilemma scenarios. Nevertheless there are several properties that heavily influence the best performing strategies, refining the properties described by R. Axelrod in light of recent and more diverse opponent populations. These are: be nice, be provocable and contrite, be a little envious, be clever, and adapt to the environment, which includes the parameters of the tournament (e.g. noise) and the population of opponents. More precisely, we find that strategies perform best when their probability of cooperation matches the total tournament population's aggregate cooperation probabilities, or

a proportion thereof in the case of noisy and probabilistically ending tournaments, and that the manner in which a strategy achieves the ideal cooperation rate is crucial. The features of high performing strategies reveal why strategies such as Tit For Tat performed historically well in tournaments and why zero-determinant strategies typically do not fare well in tournament settings.

---

The differences between the Chapter and the manuscript include ....

## 1.1 Introduction

Conceptualising strategies and understanding the best way of playing the game has been of interest to the scientific community since the formulation of the game in 1950 [22]. Following the computer tournaments of Axelrod in the 1980's [12, 13], a strategy's performance in a round robin computer tournament became a common evaluation technique for newly designed strategies. Many tournaments have followed Axelrod's [18, 17, 46, **?**, 47, 27] and today more hundreds of strategies exist in the literature.

The winner of both of Axelrod's tournaments [12, 13] was the simple strategy Tit For Tat (TFT) which cooperated on the first turn and then simply copied the previous action of its opponent, retailiating against defections with a defection, and forgiving a defection if followed by a cooperation. Axelrod concluded that the strategy's robustness was due to four properties, which he adapted into four suggestions on doing well in an IPD:

- Do not be envious by striving for a payoff larger than the opponent's payoff
- Be "nice"; Do not be the first to defect
- Reciprocate both cooperation and defection; Be provocable to retaliation and forgiveness
- Do not be too clever by scheming to exploit the opponent

As a result of the strategy's strong performance in both tournaments, and moreover in a series of evolutionary experiments [14], TFT was often claimed to be the most robust basic strategy in the IPD.

There are strategies which have built upon TFT and the reciprocity based approach. In [17] Gradual was introduced which was constructed to have the same qualities as those of TFT except one: Gradual had a memory of the game since the beginning of it. Gradual recorded the number of defections by the opponent and punished them with a growing number of defections. It would then enter a calming state in which it would cooperates for two rounds. A strategy with the same intuition as Gradual is Adaptive Tit for Tat [48]. Adaptive Tit for Tat maintains a continually updated estimate of the opponent's behaviour, and uses this estimate to condition its future actions.

Other works have built upon the limitations of TFT, and others have shown that suggestions made by Axelrod did not necessarily apply in alternative environmental settings. In [18, 21, 38, 45] it was shown that TFT suffered in environments with noise. This was mainly due to the strategy's lack of generosity and contrition. Since TFT immediately punishes a defection, in a noisy environment it can get stuck in a repeated cycle of defections and cooperations. Some new strategies, more robust in tournaments with noise, were soon introduced, including Nice and Forgiving [18], Generous Tit For Tat [41], and Pavlov (aka Win Stay Lose Shift) [40], as well as later variants such as OmegaTFT [29]. Introduction of new strategies is often accompanied by a claim that the new strategy is the best known despite only being tested against a small number of opponents, or specific classes of opponents not necessarily representative of all possible or all published strategies. The lack of testing against formally defined strategies and tournament winners is understandable given the effort required to implement [1] the hundreds of published IPD strategies, yet calls into question any claims of superiority or robustness of newly introduced strategies.

A set of envious IPD strategies were introduced called zero-determinant strategies (ZDs) in [43]. By forcing a linear relationship between stationary payoffs ZDs can ensure that they will never receive less than their memory-one opponents. While ZDs were introduced with a small tournament in which some were reportedly successful [47], this result has not generally held in future work. In [27] a series of strategies trained using reinforcement learning were introduced, and a tournament containing over 200 strategies featured no ZD strategies ranked in top spots. Instead, the top ranked strategies were a set of "clever" (in the sense of Axelrod's characteristics) trained strategies based on lookup tables [15], hidden Markov models [27], and finite state automata [36]. Similarly, in [35], a set of evolutionarily-trained strategies, and a pre-selected set of known strategies, outperformed a collection of 12 ZDs.

Though only select pieces of work have been discussed, there is a broad collection of strategies in the literature, and new strategies and competitions are being published frequently [**?**]. The question, however, still remains the same: what is the best way to play the game?

Compared to other works, where typically a few selected or introduced strategies are evaluated on a small number of tournaments and/or small number of opponents, this manuscript evaluates the performance of 195 strategies in 45686 tournaments. Furthermore a large portion of the strategies used in this manuscript are drawn from the known and named strategies in IPD literature, including many previous tournament winners, in contrast to other work that may have randomly generated many essentially arbitrary strategies (typically restrained to a class such as memory-one strategies, or those of a certain structural form such as finite state machines or deterministic memory two strategies). Additionally, our tournaments come in a number of variations including standard tournaments emulating Axelrod's original tournaments, tournaments with noise, probabilistic match length, and both noise and probabilistic match length. This diversity of strategies and tournament types yields new insights and tests earlier claims in alternative settings against known powerful strategies.

The later part of the paper evaluates the impact of features on the performance of the strategies using modern machine learning techniques. These features include measures regarding a strategy's behaviour and measures regarding the tournaments. The outcomes of our work reinforce the discussion started by Axelrod, and it concludes that the properties of a successful strategy in the IPD are:

- ~~Do not be envious~~ Be a little bit envious
- Be "nice"; Do not be the first to defect

[1]Implementing prior strategies faithfully is often extremely difficult or impossible due to insufficient descriptions and lack of published implementations or code.

- Reciprocate both cooperation and defection; Be provocable and forgiving
- ~~Do not be too clever~~ It's ok to be clever
- Adapt to the environment; Adjust to the mean population cooperation

The Chapter is structured as follows:

- The different tournament types as well as the data collection, made possible due to an open source library called Axelrod-Python (APL), are covered in Section 1.3.
- Section 1.4, focuses on the best performing strategies for each type of tournament and overall.
- Section 1.5, explores the traits which contribute to good performance

This manuscripts uses several parameters. These are introduced in the following sections, however, the full set of parameters and their definitions are given in Appendix **??**.

## 1.2 Data collection

## 1.3 Data collection

The data set generated for this manuscript was created with APL version 3.0.0. APL allows for different types of IPD computer tournaments to be simulated and contains a large list of strategies. Most of these are strategies described in the literature with a few exceptions of strategies that have been contributed specifically to the package. This paper makes use of 195 strategies implemented in version 3.0.0. A list of the strategies is given in the Appendix A.1. Although APL features several tournament types, this work considers standard, noisy, probabilistic ending, and noisy probabilistic ending tournaments.

**Standard tournaments** are tournaments similar to that of Axelrod's well-known tournaments [12]. There are $N$ strategies which all play an iterated game of $n$ number of turns against each other. Note that self-interactions are not included. Similarly, **noisy tournaments** have $N$ strategies and $n$ number of turns, but at each turn there is a probability $p_n$ that a player's action will be flipped. **Probabilistic ending tournaments**, are of size $N$ and after each turn a match between strategies ends with a given probability $p_e$. Finally, **noisy probabilistic ending** tournaments have both a noise probability $p_n$ and an ending probability $p_e$. For smoothing the simulated results a tournament is repeated for $k$ number of times. This was allowed to vary in order to evaluate the effect of smoothing. The winner of each tournament is based on the average score a strategy achieved and not by the number of wins.

The process of collecting tournament results is described by Algorithm 1. For each trial a random size $N$ is selected, and from the 195 strategies a random list of $N$ strategies is chosen. For the given list of strategies a standard, a noisy, a probabilistic ending and a noisy probabilistic ending tournament are performed and repeated $k$ times. The parameters for the tournaments, as well as the number of repetitions, are selected once for each trial. The parameters and their respective minimum and maximum values are given by Table 1.1.

The source code for the data collection, as well as the source code for the analysis, which will be discussed in the following sections, have been written following best practices [5, 19] and is

| parameter | parameter explanation | min value | max value |
|---|---|---|---|
| $N$ | number of strategies | 3 | 195 |
| $k$ | number of repetitions | 10 | 100 |
| $n$ | number of turns | 1 | 200 |
| $p_n$ | probability of flipping action at each turn | 0 | 1 |
| $p_e$ | probability of match ending in the next turn | 0 | 1 |

Table 1.1: Data collection; parameters' values

available here.

**Algorithm 1:** Data collection Algorithm

**foreach** *seed* $\in [0, 11420]$ **do**
  $N \leftarrow$ randomly select integer $\in [N_{min}, N_{max}]$;
  players $\leftarrow$ randomly select $N$ players;
  $k \leftarrow$ randomly select integer $\in [k_{min}, k_{max}]$;
  $n \leftarrow$ randomly select integer $\in [n_{min}, n_{max}]$;
  $p_n \leftarrow$ randomly select float $\in [p_{n\,min}, p_{n\,max}]$;
  $p_e \leftarrow$ randomly select float $\in [p_{e\,min}, p_{e\,max}]$;

  result standard $\leftarrow$ Axelrod.tournament(players, $n, k$);
  result noisy $\leftarrow$ Axelrod.tournament(players, $n, p_n, k$);
  result probabilistic ending $\leftarrow$ Axelrod.tournament(players, $p_e, k$);
  result noisy probabilistic ending $\leftarrow$ Axelrod.tournament(players, $p_n, p_e, k$);
**return** *result standard, result noisy, result probabilistic ending, result noisy probabilistic ending*;

A total of 11420 trials of Algorithm 1 have been run. For each trial the results for 4 different tournaments were collected, thus a total of 45686 (11420 × 4) tournament results have been retrieved. Each tournament outputs a result summary in the form of Table 1.2. Each strategy have participated on average in 5154 tournaments of each type. The strategy with the maximum participation in each tournament type is Inverse Punisher with 5639 entries. The strategy with the minimum entries is EvolvedLookerUp 1 1 1 which was selected in 4693 trials.

A result summary (Table 1.2) has $N$ number of rows because each row contains information for each strategy that participated in the tournament. The information includes the strategy's rank, median score, the rate with which the strategy cooperated ($C_r$), its match win count, and the probability that the strategy cooperated in the opening move. Moreover, the probabilities of a strategy being in any of the four states ($CC, CD, DC, DD$), and the rate of which the strategy cooperated after each state. The **normalised rank** feature that is manually added. The rank $R$ of a given strategy can vary between 0 and $N - 1$. Thus, the normalised rank, denoted as $r$, is calculated as a strategy's rank divided by $N - 1$.

| | | | | | | Rates | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rank | Name | Median score | Cooperation rating ($C_r$) | Win | Initial C | CC | CD | DC | DD | CC to C | CD to C | DC to C | DD to C |
| 0 | EvolvedLookerUp2 2 2 | 2.97 | 0.705 | 28.0 | 1.0 | 0.639 | 0.066 | 0.189 | 0.106 | 0.836 | 0.481 | 0.568 | 0.8 |
| 1 | Evolved FSM 16 Noise 05 | 2.875 | 0.697 | 21.0 | 1.0 | 0.676 | 0.020 | 0.135 | 0.168 | 0.985 | 0.571 | 0.392 | 0.07 |
| 2 | PSO Gambler 1 1 1 | 2.874 | 0.684 | 23.0 | 1.0 | 0.651 | 0.034 | 0.152 | 0.164 | 1.000 | 0.283 | 0.000 | 0.136 |
| 3 | PSO Gambler Mem1 | 2.861 | 0.706 | 23.0 | 1.0 | 0.663 | 0.042 | 0.145 | 0.150 | 1.000 | 0.510 | 0.000 | 0.122 |
| 4 | Winner12 | 2.835 | 0.682 | 20.0 | 1.0 | 0.651 | 0.031 | 0.141 | 0.177 | 1.000 | 0.441 | 0.000 | 0.462 |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … |

Table 1.2: Output result of a single tournament.

## 1.4 Top ranked strategies

The performance of each strategy is evaluated in four tournament types, as presented in Section 1.3, followed by an evaluation of their performance over all the 45686 simulated tournaments of this work. Each strategy participated in multiple tournaments of the same type (on average 5154). For example TFT participated in a total of 5114 tournaments of each type. The strategy's normalised rank distribution in these is given in Figure 1.1. A value of $r = 0$ corresponds to a strategy winning the tournament where a value of $r = 1$ corresponds to the strategy coming last. Because of the strategies' multiple entries their performance is evaluated based on the **median normalised rank** denoted as $\bar{r}$.

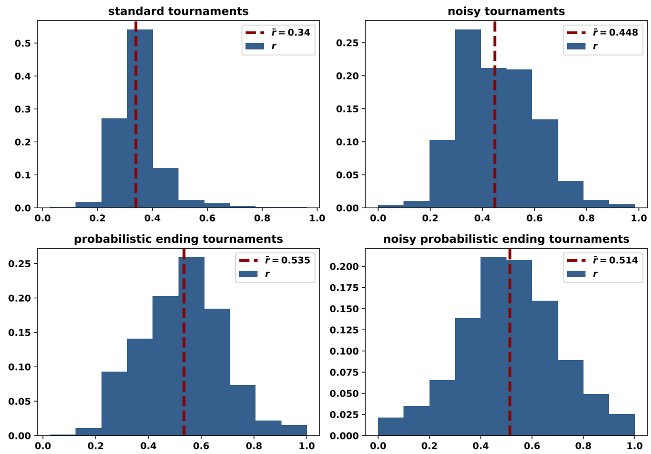
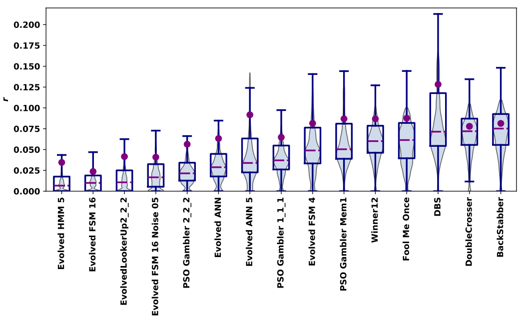



Figure 1.1: TFT's $r$ distribution in tournaments. Lower values of $r$ correspond to better performances. The best performance of the strategy has been in standard tournaments where it achieved a $\bar{r}$ of 0.34.

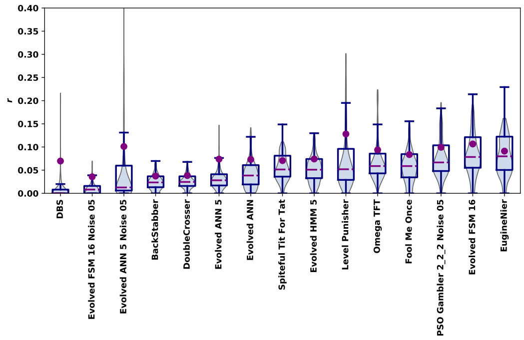
The top 15 strategies for each tournament type based on $\bar{r}$ are given in Table 1.3. The data collection process was designed such that the probabilities of noise and ending of the match varied between 0 and 1. However, commonly used values of these probabilities are values are often less than 0.1. Thus, Table 1.3 also includes the top 15 strategies in noisy tournaments with $p_n < 0.1$ and probabilistic ending tournaments with $p_e < 0.1$.

The $r$ distributions for the top ranked strategies of Table 1.3 are given by Figure 1.2.

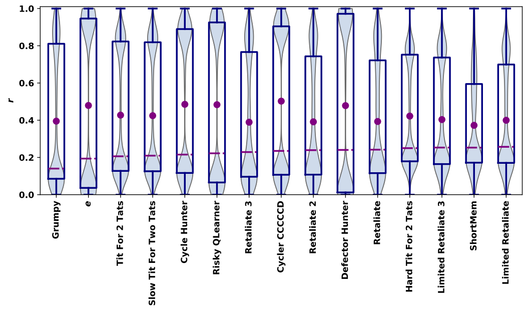
In standard tournaments 10 out of the 15 top strategies are introduced in [27]. These are strategies based on finite state automata (FSM), hidden markov models (HMM), artificial neural networks (ANN), lookup tables (LookerUp) and stochastic lookup tables (Gambler) that have been trained using reinforcement learning algorithms (evolutionary and particle swarm algorithms). They have been trained to perform well against the strategies in APL in a standard



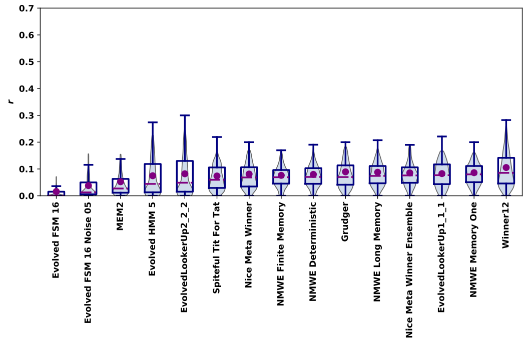

(a) $r$ distributions of top 15 strategies in standard tournaments.



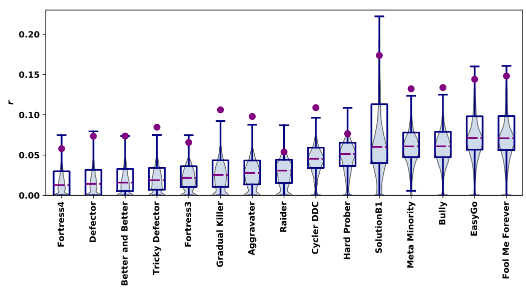

(b) $r$ distributions of top 15 strategies in noisy tournaments with $p_n < 0.1$.



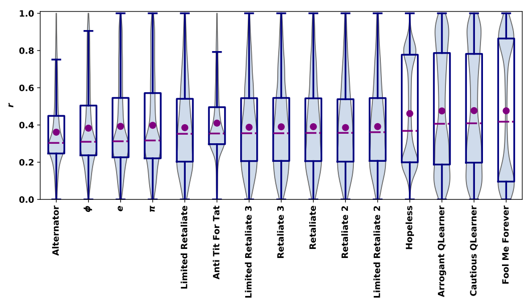

(c) $r$ distributions of top 15 strategies in noisy tournaments.




(d) $r$ distributions of top 15 strategies in 1139 probabilistic ending tournaments with $p_e < 0.1$.




(e) $r$ distributions of top 15 strategies in probabilistic ending tournaments.




(f) $r$ distributions of top 15 strategies in noisy

| | Standard | | Noisy | | Noisy ($p_n < 0.1$) | | Probabilistic ending | | Probabilistic ending ($p_e < 0.1$) | | Noisy probabilistic ending | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Name | $\bar{r}$ | Name | $\bar{r}$ | Name | $\bar{r}$ | Name | $\bar{r}$ | Name | $\bar{r}$ | Name | $\bar{r}$ |
| 0 | Evolved HMM 5 | 0.007 | Grumpy | 0.140 | DBS | 0.000 | Fortress4 | 0.013 | Evolved FSM 16 | 0.000 | Alternator | 0.304 |
| 1 | Evolved FSM 16 | 0.010 | $e$ | 0.194 | Evolved FSM 16 Noise 05 | 0.008 | Defector | 0.014 | Evolved FSM 16 Noise 05 | 0.013 | $\phi$ | 0.310 |
| 2 | EvolvedLookerUp2 2 2 | 0.011 | Tit For 2 Tats | 0.206 | Evolved ANN 5 Noise 05 | 0.013 | Better and Better | 0.016 | MEM2 | 0.027 | $e$ | 0.312 |
| 3 | Evolved FSM 16 Noise 05 | 0.017 | Slow Tit For Two Tats | 0.210 | BackStabber | 0.024 | Tricky Defector | 0.019 | Evolved HMM 5 | 0.044 | $\pi$ | 0.317 |
| 4 | PSO Gambler 2 2 2 | 0.021 | Cycle Hunter | 0.215 | DoubleCrosser | 0.025 | Fortress3 | 0.022 | EvolvedLookerUp2 2 2 | 0.049 | Limited Retaliate | 0.353 |
| 5 | Evolved ANN | 0.029 | Risky QLearner | 0.222 | Evolved ANN 5 | 0.028 | Gradual Killer | 0.025 | Spiteful Tit For Tat | 0.060 | Anti Tit For Tat | 0.354 |
| 6 | Evolved ANN 5 | 0.034 | Retaliate 3 | 0.229 | Evolved ANN | 0.038 | Aggravater | 0.028 | Nice Meta Winner | 0.068 | Limited Retaliate 3 | 0.356 |
| 7 | PSO Gambler 1 1 1 | 0.037 | Cycler CCCCCD | 0.235 | Spiteful Tit For Tat | 0.051 | Raider | 0.031 | NMWE Finite Memory | 0.069 | Retaliate 3 | 0.356 |
| 8 | Evolved FSM 4 | 0.049 | Retaliate 2 | 0.239 | Evolved HMM 5 | 0.051 | Cycler DDC | 0.045 | NMWE Deterministic | 0.070 | Retaliate | 0.357 |
| 9 | PSO Gambler Mem1 | 0.050 | Defector Hunter | 0.240 | Level Punisher | 0.052 | Hard Prober | 0.051 | Grudger | 0.070 | Retaliate 2 | 0.358 |
| 10 | Winner12 | 0.060 | Retaliate | 0.242 | Omega TFT | 0.059 | SolutionB1 | 0.060 | NMWE Long Memory | 0.074 | Limited Retaliate 2 | 0.361 |
| 11 | Fool Me Once | 0.061 | Hard Tit For 2 Tats | 0.250 | Fool Me Once | 0.059 | Meta Minority | 0.061 | Nice Meta Winner Ensemble | 0.076 | Hopeless | 0.368 |
| 12 | DBS | 0.071 | Limited Retaliate 3 | 0.253 | PSO Gambler 2 2 2 Noise 05 | 0.067 | Bully | 0.061 | EvolvedLookerUp1 1 1 | 0.077 | Arrogant QLearner | 0.407 |
| 13 | DoubleCrosser | 0.072 | ShortMem | 0.253 | Evolved FSM 16 | 0.078 | EasyGo | 0.071 | NMWE Memory One | 0.080 | Cautious QLearner | 0.409 |
| 14 | BackStabber | 0.075 | Limited Retaliate | 0.257 | EugineNier | 0.080 | Fool Me Forever | 0.071 | Winner12 | 0.085 | Fool Me Forever | 0.418 |

Table 1.3: Top performances for each tournament type based on $\bar{r}$. The results of each type are based on 11420 unique tournaments of each type. The results for noisy tournaments with $p_n < 0.1$ are based on 1151 tournaments, and for probabilistic ending tournaments with $p_e < 0.1$ on 1139. The top ranks indicate that trained strategies perform well in a variety of environments, but so do simple deterministic strategies. The normalised medians are close to 0 for most environments, except environments with noise not restricted to 0.1 regardless the number of turns. Noisy and noisy probabilistic ending tournaments have the highest medians. This implies that strategies from the collection of this work do not perform well in environments with high values of noise.

tournament, thus their performance in the specific setting was anticipated. DoubleCrosser, BackStabber and Fool Me Once, are strategies not from the literature but from the APL. DoubleCrosser is an extension of BackStabber and both strategies make use of the number of turns because they are set to defect on the last two rounds. It should be noted that these strategies can be characterised as "cheaters" because the source code of the strategies allows them to know the number of turns in a match (unless the match has a probabilistic ending). These strategies were expected to not perform as well in tournaments where the number of turns is not specified. Finally, Winner 12 [35] and DBS [11] are both from the literature. DBS is a strategy specifically designed for noisy environments, however, it ranks highly in standard tournaments as well. Similarly the fourth ranked player, Evolved FSM 16 Noise 05, was trained for noisy tournaments yet performs well in standard tournaments. Figure 1.2a shows that these strategies typically perform well in any standard tournament in which they participate.

In the case of noisy tournaments with $p_n < 0.1$ the top performed strategies include strategies specifically designed for noisy tournaments. These are DBS, Evolved FSM 16 Noise 05, Evolved ANN 5 Noise 05, PSO Gambler 2 2 2 Noise 05 and Omega Tit For Tat [29]. Omega TFT, another strategy designed to break the deadlocking cycles of CD and DC that TFT can fall into in noisy environments, places 10th. The rest of the top ranks are occupied by strategies which performed well in standard tournaments and deterministic strategies such as Spiteful Tit For Tat [1], Level Punisher [4], Eugine Nier [42]. Similarly to standard tournaments, the successful strategies in this given setting performed well overall in the tournaments they participated in, Figure 1.2b.

In comparison, the performance of the top ranked strategies in noisy environments when $p_n \in [0, 1]$ is bimodal. The top strategies include strategies which decide their actions based on the cooperation to defection ratio, such as ShortMem [20], Grumpy [3] and e [3], and the Retaliate strategies which are designed to defect if the opponent has tricked them more often than a given percentage of the times that they have done the same. The bimodality of the $r$ distributions is explained by Figure 1.3 which demonstrates that the top 6 strategies were highly ranked due to the their performance in tournaments with $p_n > 0.5$, and that in tournaments with a noise probability lower than 0.5 they performed poorly. At a noisy level of 0.5 or greater, mostly cooperative strategies become mostly defectors and vice versa.

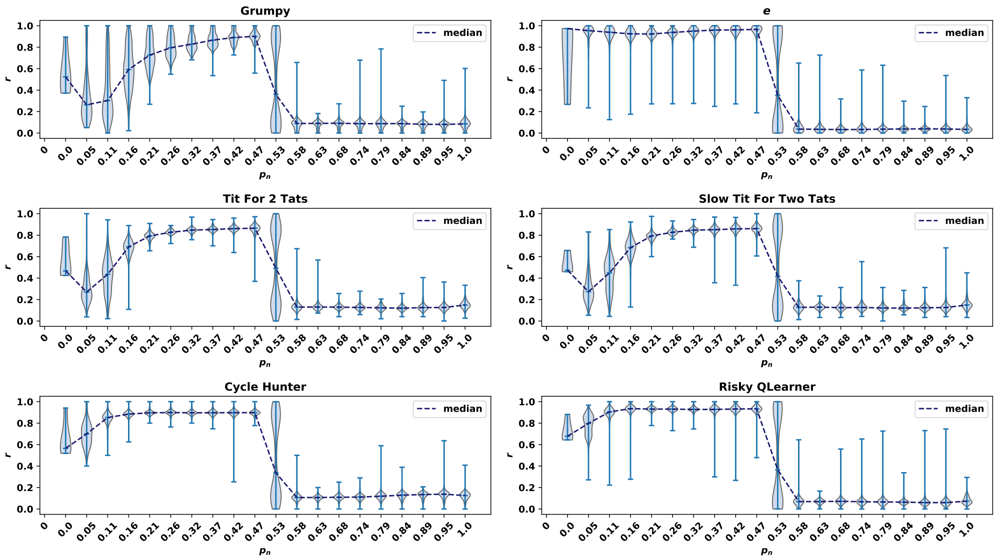



Figure 1.3: $r$ distributions for top 6 strategies in noisy tournaments over the probability of noisy ($p_n$).

The new entrants to the most effective strategies list in probabilistic ending tournaments with $p_e < 0.1$ are a series of Meta strategies, trained strategies which performed well in standard tournaments, and Grudger [3] and Spiteful Tit for Tat [1]. The Meta strategies [3] create a team of strategies and play as an ensemble or some other combination of their team members. Figure 1.2d indicates that these strategies performed well in any probabilistic ending tournament they competed in.

In probabilistic ending tournaments with $p_e \in [0, 1]$ the top ranks are mostly occupied by defecting strategies such as Better and Better, Gradual Killer, Hard Prober (all from [3]), Bully (Reverse Tit For Tat) [39] and Defector, and a series of strategies based on finite state automata introduced by Daniel Ashlock and Wendy Ashlock; Fortress 3, Fortress 4 (both introduced in [9]), Raider [10] and Solution B1 [10]. The success of defecting strategies in probabilistic ending tournaments is due to larger values of $p_e$ which lead to shorter matches (the expected number of rounds is $1/p_e$), so the impact of the PD being iterated is subdued. This is captured by the Folk Theorem [24] as defecting strategies do better when the likelihood of the game ending in the next turn increases. This is demonstrated by Figure 1.4, which gives the distributions of $r$ for the top 6 strategies in probabilistic ending tournaments over $p_e$.

The top performances in tournaments with both noise and a probabilistic ending and the top performances over the entire data set have the largest median values compared to the top rank strategies of the other tournament types, Figure 1.2f and Figure 1.5. The $\bar{r}$ for the top strategy is approximately at 0.3, indicating that the most successful strategy can on average just place at the top 30% of the competition.

On the whole, the analysis of this manuscript has shown that:

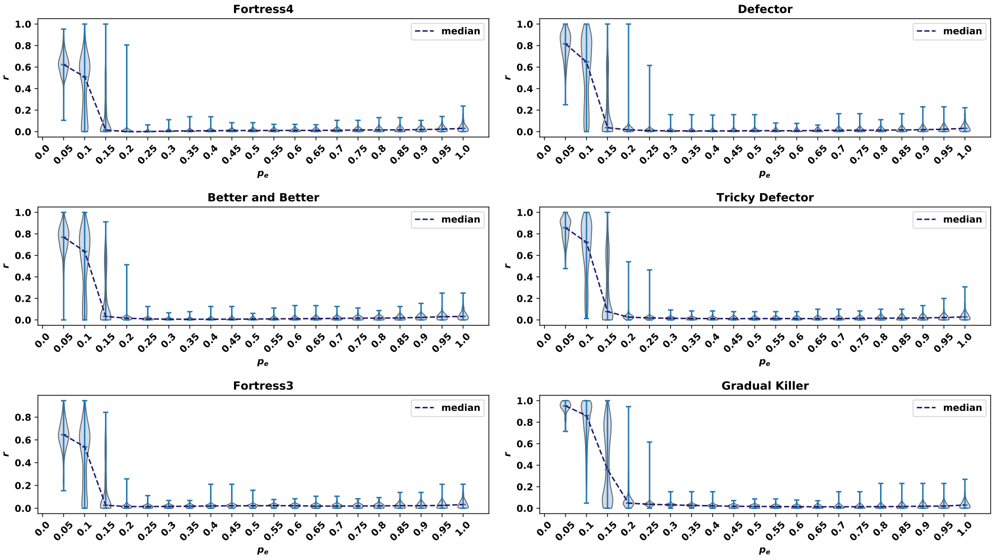



Figure 1.4: $r$ distributions for top 6 strategies in probabilistic ending tournaments over $p_e$. The 6 strategies start of with a high median rank, however, their ranked decreased as the the probability of the game ending increased and at the point of $p_e = 0.1$.

| Name | $\bar{r}$ |
|---|---|
| Limited Retaliate 3 | 0.286 |
| Retaliate 3 | 0.296 |
| Retaliate 2 | 0.302 |
| Limited Retaliate 2 | 0.303 |
| Limited Retaliate | 0.310 |
| Retaliate | 0.317 |
| BackStabber | 0.324 |
| DoubleCrosser | 0.331 |
| Nice Meta Winner | 0.349 |
| PSO Gambler 2 2 2 Noise 05 | 0.351 |
| Grudger | 0.352 |
| Evolved HMM 5 | 0.357 |
| NMWE Memory One | 0.357 |
| Nice Meta Winner Ensemble | 0.359 |
| Forgetful Fool Me Once | 0.359 |

Table 1.4: Top performances over all the tournaments. The top ranks include strategies that have been previously mentioned. The set of Retaliate strategies occupy the top spots followed by BackStabber and DoubleCrosser. The distributions of the Retaliate strategies have no statistical difference. PSO Gambler and Evolved HMM 5 are trained strategies introduced in [27] and Nice Meta Winner and NMWE Memory One are strategies based on teams. Grudger is a strategy from Axelrod's original tournament and Forgetful Fool Me Once is based on the same approach as Grudger.

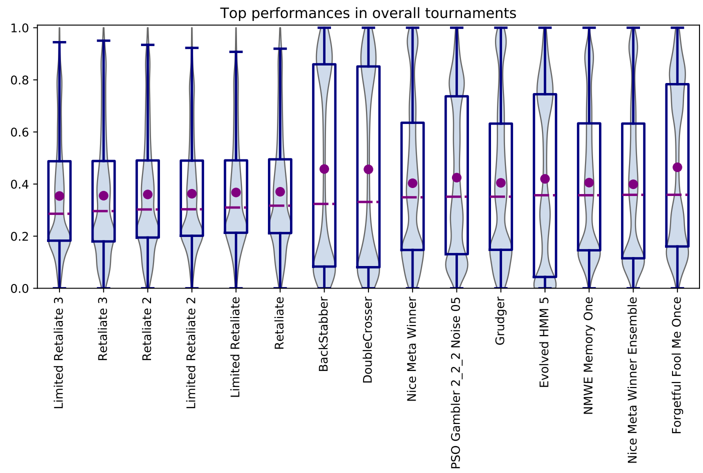



Figure 1.5: $r$ distributions for best performed strategies in the data set [26]. A lower value of $\bar{r}$ corresponds to a more successful performance.

- In standard tournaments the dominating strategies were strategies that had been trained using reinforcement learning techniques.
- In noisy environments where the noise probability strictly less than 0.1 was considered, the successful strategies were strategies specifically designed for noisy environments.
- In probabilistic ending tournaments most of the highly ranked strategies were defecting strategies and trained finite state automata, all by the authors of [9, 10]. These strategies ranked high due to their performance in tournaments where the probability of the game ending after each turn was bigger than 0.1.
- In probabilistic tournaments with $p_e$ less than 0.1 the highly ranked strategies were strategies based on the behaviour of others.
- From the collection of strategies considered here, no strategy can be consistently successful in noisy environments, except if the value of noise is constrained to less than a 0.1.

Though there is not a single strategy that repeatably outranks all others in any of the distinct tournament types, or even across the tournaments type, there are specific types of strategies have been repeatably ranked in the top ranks. These have been strategies that have been trained, strategies that retailiate, and strategies that would adapt their behaviour based on preassigned rules to achieve the highest outcome. These results contradict some of Axelrod's suggestions, and more specifically, the suggestions 'Do not be clever' and 'Do not be envious'. The features and properties contributing a strategy's success are further explored in Section 1.5.

## 1.5 Evaluation of performance

Now we examine performance of the strategies based on features of strategies described in Table 1.5. These features are measures regarding a strategy's behaviour from the tournaments the strategies competed in as well as intrinsic properties such as whether a strategy is deterministic or stochastic.

| feature | feature explanation | source | value type | min value | max value |
|---|---|---|---|---|---|
| stochastic | If a strategy is stochastic | strategy classifier from APL | boolean | Na | Na |
| makes use of game | If a strategy makes used of the game information | strategy classifier from APL | boolean | Na | Na |
| makes use of length | If a strategy makes used of the number of turns | strategy classifier from APL | boolean | Na | Na |
| memory usage | The memory size of a strategy divided by the number of turns | memory size from APL | float | 0 | 1 |
| SSE | A measure of how far a strategy is from ZD behaviour | method described in [30] | float | 0 | 1 |
| max cooperating rate ($C_{\max}$) | The biggest cooperating rate in a given tournament | result summary | float | 0 | 1 |
| min cooperating rate ($C_{\min}$) | The smallest cooperating rate in a given tournament | result summary | float | 0 | 1 |
| median cooperating rate ($C_{\text{median}}$) | The median cooperating rate in a given tournament | result summary | float | 0 | 1 |
| mean cooperating rate ($C_{\text{mean}}$) | The mean cooperating rate in a given tournament | result summary | float | 0 | 1 |
| $C_r$ / $C_{\max}$ | A strategy's cooperating rate divided by the maximum | result summary | float | 0 | 1 |
| $C_{\min}$ / $C_r$ | A strategy's cooperating rate divided by the minimum | result summary | float | 0 | 1 |
| $C_r$ / $C_{\text{median}}$ | A strategy's cooperating rate divided by the median | result summary | float | 0 | 1 |
| $C_r$ / $C_{\text{mean}}$ | A strategy's cooperating rate divided by the mean | result summary | float | 0 | 1 |
| $C_r$ | The cooperating ratio of a strategy | result summary | float | 0 | 1 |
| $CC$ to $C$ rate | The probability a strategy will cooperate after a mutual cooperation | result summary | float | 0 | 1 |
| $CD$ to $C$ rate | The probability a strategy will cooperate after being betrayed by the opponent | result summary | float | 0 | 1 |
| $DC$ to $C$ rate | The probability a strategy will cooperate after betraying the opponent | result summary | float | 0 | 1 |
| $DD$ to $C$ rate | The probability a strategy will cooperate after a mutual defection | result summary | float | 0 | 1 |
| $p_n$ | The probability of a player's action being flip at each interaction | trial summary | float | 0 | 1 |
| $n$ | The number of turns | trial summary | integer | 1 | 200 |
| $p_e$ | The probability of a match ending in the next turn | trial summary | float | 0 | 1 |
| $N$ | The number of strategies in the tournament | trial summary | integer | 3 | 195 |
| $k$ | The number of repetitions of a given tournament | trial summary | integer | 10 | 100 |

Table 1.5: The features which are included in the performance evaluation analysis. Stochastic, makes use of length and makes use of game are APL classifiers that determine whether a strategy is stochastic or deterministic, whether it makes use of the number of turns or the game's payoffs. The memory usage is calculated as the number of turns the strategy considers to make an action (which is specified in the APL) divided by the number of turns. The SSE (introduced in [30]) shows how close a strategy is to behaving as a ZDs, and subsequently, in an extortionate way. The method identifies the ZDs closest to a given strategy and calculates the algebraic distance between them, defined as SSE. A SSE value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is behaving a ZDs. The rest of the features considered are the $CC$ to $C$, $CD$ to $C$, $DC$ to $C$, and $DD$ to $C$ rates as well as cooperating ratio of a strategy, the minimum ($C_{min}$), maximum ($C_{max}$), mean ($C_{mean}$) and median ($C_{median}$) cooperating ratios of each tournament.

The memory usage of strategies with an infinite memory size, for example Evolved FSM 16 Noise 05, is equal to 1. Otherwise the memory usage is the number of rounds of play used by the strategy divided by the number of turns in each match. For example, Winner12 uses the previous two rounds of play, and if participating in a participated in a tournament where $n$ was 100 the memory usage would be 2/100. Note that for tournaments with a probabilistic ending the number of turns was not collected, so the memory usage feature is not used for probabilistic ending tournaments.

The correlation coefficients between the features of Table 1.5 the median score and the median normalised rank are given by Table 1.6. The correlation coefficients between all features of Table 1.5 have been calculated and a graphical representation can be found in the Appendix B.

| | Standard | | Noisy | | Probabilistic ending | | Noisy probabilistic ending | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $r$ | median score | $r$ | median score | $r$ | median score | $r$ | median score | $r$ | median score |
| $CC$ to $C$ rate | -0.501 | 0.501 | 0.414 | -0.504 | 0.408 | -0.323 | 0.260 | 0.022 | -0.501 | 0.501 |
| $CD$ to $C$ rate | 0.226 | -0.199 | 0.456 | -0.330 | 0.320 | -0.017 | 0.205 | -0.220 | 0.226 | -0.199 |
| $C_r$ | -0.323 | 0.384 | 0.711 | -0.678 | 0.714 | -0.832 | 0.579 | -0.135 | -0.323 | 0.384 |
| $C_r$ / $C_{max}$ | -0.323 | 0.381 | 0.616 | -0.551 | 0.714 | -0.833 | 0.536 | -0.116 | -0.323 | 0.381 |
| $C_r$ / $C_{mean}$ | -0.331 | 0.358 | 0.731 | -0.740 | 0.721 | -0.861 | 0.649 | -0.621 | -0.331 | 0.358 |
| $C_r$ / $C_{median}$ | -0.331 | 0.353 | 0.652 | -0.669 | 0.712 | -0.852 | 0.330 | -0.466 | -0.331 | 0.353 |
| $C_r$ / $C_{min}$ | 0.109 | -0.080 | -0.358 | 0.250 | -0.134 | 0.150 | -0.368 | 0.113 | 0.109 | -0.080 |
| $C_{max}$ | -0.000 | 0.049 | 0.000 | 0.023 | -0.000 | 0.046 | 0.000 | -0.004 | -0.000 | 0.049 |
| $C_{mean}$ | -0.000 | 0.229 | -0.000 | 0.271 | 0.000 | 0.200 | 0.000 | 0.690 | -0.000 | 0.229 |
| $C_{median}$ | 0.000 | 0.209 | -0.000 | 0.240 | -0.000 | 0.187 | -0.000 | 0.673 | 0.000 | 0.209 |
| $C_{min}$ | 0.000 | 0.084 | 0.000 | -0.017 | -0.000 | 0.007 | -0.000 | 0.041 | 0.000 | 0.084 |
| $DC$ to $C$ rate | 0.127 | -0.100 | 0.509 | -0.504 | -0.018 | 0.033 | 0.341 | -0.016 | 0.127 | -0.100 |
| $DD$ to $C$ rate | 0.412 | -0.396 | 0.533 | -0.436 | -0.103 | 0.176 | 0.378 | -0.263 | 0.412 | -0.396 |
| $N$ | 0.000 | -0.009 | -0.000 | 0.002 | -0.000 | 0.003 | -0.000 | 0.001 | 0.000 | -0.009 |
| $k$ | 0.000 | -0.002 | -0.000 | 0.003 | -0.000 | 0.001 | -0.000 | -0.008 | 0.000 | -0.002 |
| $n$ | 0.000 | -0.125 | -0.000 | -0.024 | - | - | - | - | 0.000 | -0.125 |
| $p_e$ | - | - | - | - | 0.000 | 0.165 | 0.000 | -0.058 | -0.001 | 0.001 |
| $p_n$ | - | - | -0.000 | 0.207 | - | - | -0.000 | -0.650 | 0.002 | -0.000 |
| Make use of game | -0.003 | -0.022 | 0.025 | -0.082 | -0.053 | -0.108 | 0.013 | -0.016 | -0.003 | -0.022 |
| Make use of length | -0.158 | 0.124 | 0.005 | -0.123 | -0.025 | -0.090 | 0.014 | -0.016 | -0.154 | 0.117 |
| SSE | 0.473 | -0.452 | 0.463 | -0.337 | -0.156 | 0.223 | 0.305 | -0.259 | 0.473 | -0.452 |
| memory usage | -0.082 | 0.095 | -0.007 | -0.017 | - | - | - | - | -0.084 | 0.095 |
| stochastic | 0.006 | -0.024 | 0.022 | -0.026 | 0.002 | -0.130 | 0.021 | -0.013 | 0.006 | -0.024 |

Table 1.6: Correlations table between the features of Table 1.5 the normalised rank and the median score.

In standard tournaments the features $CC$ to $C$, $C_r$, $C_r/C_{\max}$ and the cooperating ratio compared to $C_{\text{median}}$ and $C_{\text{mean}}$ have a moderate negative effect on the normalised rank, and a moderate positive on the median score. The SSE error and the $DD$ to $C$ have the opposite effects. Thus, in standard tournaments behaving cooperatively corresponds to a more successful performance. Even though being nice pays off, that's not true against defective strategies. Being more cooperative after a mutual defection is associated to lesser overall success in terms of normalised rank. Figure 1.6 confirms that the winners of standard tournaments always cooperate after a mutual cooperation and almost always defect after a mutual defection.

Compared to standard tournaments, in both noisy and in probabilistic ending tournaments the higher the rates of cooperation the lower a strategy's success and median score. A strategy would want to cooperate less than both the mean and median cooperator in such settings. In probabilistic ending tournaments the correlation coefficients have larger values, indicating a stronger effect. Thus a strategy will be punished more by its cooperative behaviour in prob-

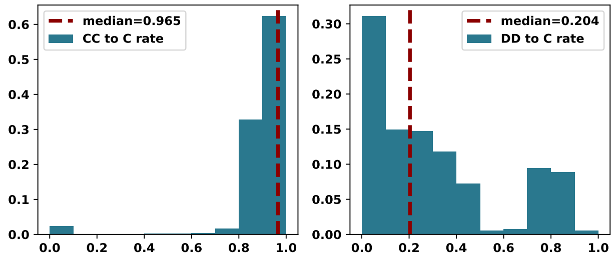



Figure 1.6: Distributions of $CC$ to $C$ and $DD$ to $C$ for the winners in standard tournaments.

abilistic ending environments, supporting the results of Section 1.5 as well. The distributions of the $C_r$ of the winners in both tournaments are given by Figure 1.7. It confirms that the winners in noisy tournaments cooperated less than 35% of the time and in probabilistic ending tournaments less than 10%. In noisy probabilistic ending tournaments and over all the tournaments' results, the only features that had a moderate effect are $C_r/C_{\text{mean}}, C_r/C_{\text{max}}$ and $C_r$. In such environments cooperative behaviour appears to be punished less than in noisy and probabilistic ending tournaments.

Moreover, the manner in which a strategy achieves a given cooperation rate relative to the tournament population average is important. Playing a strategy that randomly cooperates with $C_{\text{mean}}$ is unlikely to be as effective as it will randomly have cooperations matched with defections. In contrast, TFT naturally achieves a cooperation rate near $C_{\text{mean}}$ by virtue of copying its opponent's last move while also minimizing instances where it is exploited by an opponent (cooperating while the opponent defects), at least in non-noisy tournaments. [2] TFT forces the opponent to pay back the (C, D) round with a (D, C) round before returning to mutual cooperation. This explains why TFT performed well in Axelrod's original tournaments as most strategies submitted to those tournaments were typically cooperative and relatively few strategies used an easily exploitable pattern. TFT does not appear in the top ranks in these tournaments because it is too nice. Strategies like Grudger will always defect after a fixed number of opponent defections, which allows them to effectively exploit strategies like Alternator or stochastic strategies that have a non-zero chance of cooperating after mutual defection, which TFT will not do. Moreover in a noisy environment these strategies will naturally tend toward always defecting, leading them to exploit strategies like Cooperator. In such a scenario the noisy environment effectively voids Axelrod's rule to be nice, allowing strategies to attempt exploitation, whereas in a noise-free environment, exploitation is risky because several strategies exhibit a Grudger-like behavior, reducing the overall value in attempting to exploit strategies like Cooperator.

Similarly, these results suggest an explanation regarding the intuitively unexpected effectiveness of memory one strategies historically. Given that among the important features associated with success are the relative cooperation rate to the population average and the four memory-one probabilities of cooperating conditional on the previous round of play, all five features can be optimized by a memory one strategy such as TFT. Usage of more history becomes valuable when there are exploitable opponent patterns, indicated by the importance of SSE as a feature, that the first-approximation provided by a memory one strategy is no longer sufficient.

[2]This also explains why Tit For N Tats does not fare well – it fails to achieve the proper cooperation ratio.

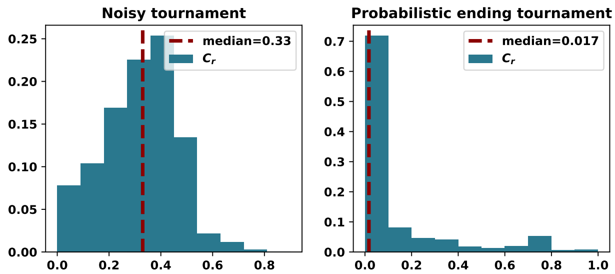



Figure 1.7: $C_r$ distributions of the winners in noisy and in probabilistic ending tournaments.

A multivariate linear regression has been fitted to model the relationship between the features and the normalised rank. Based on the graphical representation of the correlation matrices given in Appendix B several of the features are highly correlated. Highly correlated features have been removed before fitting the linear regression model. The features included are given by Table 1.7 alongside their corresponding $p$ values in the distinct tournaments and their regression coefficients.

| | Standard | | Noisy | | Probabilistic ending | | Noisy probabilistic ending | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $R$ adjusted: 0.541 | | $R$ adjusted: 0.639 | | $R$ adjusted: 0.587 | | $R$ adjusted: 0.577 | | $R$ adjusted: 0.242 | |
| | Coefficient | $p$-value | Coefficient | $p$-value | Coefficient | $p$-value | Coefficient | $p$-value | Coefficient | $p$-value |
| $CC$ to $C$ rate | -0.042 | 0.000 | -0.007 | 0.000 | 0.017 | 0.000 | 0.111 | 0.0 | -0.099 | 0.0 |
| $CD$ to $C$ rate | 0.297 | 0.000 | -0.068 | 0.000 | 0.182 | 0.000 | 0.023 | 0.0 | 0.129 | 0.0 |
| $C_r$ / $C_{max}$ | - | - | 1.856 | 0.000 | - | - | 1.256 | 0.0 | - | - |
| $C_r$ / $C_{mean}$ | -0.468 | 0.000 | -0.577 | 0.000 | 0.525 | 0.000 | -0.120 | 0.0 | 0.300 | 0.0 |
| $C_{max}$ | -0.071 | 0.000 | - | - | -0.022 | 0.391 | 1.130 | 0.0 | - | - |
| $C_{mean}$ | 0.118 | 0.000 | -2.558 | 0.000 | -0.023 | 0.001 | -1.489 | 0.0 | - | - |
| $C_{min}$ | -0.161 | 0.000 | -1.179 | 0.000 | -0.170 | 0.000 | - | - | - | - |
| $C_{min}$ / $C_r$ | 0.057 | 0.000 | -0.320 | 0.000 | 0.125 | 0.000 | - | - | -0.103 | 0.0 |
| $DC$ to $C$ rate | 0.198 | 0.000 | 0.040 | 0.000 | -0.030 | 0.000 | 0.022 | 0.0 | 0.064 | 0.0 |
| $k$ | 0.000 | 0.319 | 0.000 | 0.020 | 0.000 | 0.002 | 0.000 | 0.0 | - | - |
| $n$ | 0.000 | 0.000 | - | - | - | - | - | - | - | - |
| $p_e$ | - | - | - | - | 0.000 | 0.847 | -0.083 | 0.0 | - | - |
| $p_n$ | - | - | -0.048 | 0.000 | - | - | - | - | - | - |
| SSE | 0.258 | 0.000 | 0.153 | 0.000 | -0.041 | 0.000 | 0.100 | 0.0 | 0.056 | 0.0 |
| constant | 0.697 | 0.000 | 1.522 | 0.000 | -0.057 | 0.019 | -0.472 | 0.0 | 0.178 | 0.0 |
| memory usage | -0.010 | 0.000 | -0.000 | 0.035 | - | - | - | - | - | - |

Table 1.7: Results of multivariate linear regressions with $r$ as the dependent variable. $R$ squared is reported for each model.

A multivariate linear regression has also be fitted on the median score. The coefficients and $p$ values of the features can be found in Appendix **??**. The results of the two methods are in agreement.

The feature $C_r/C_{\text{mean}}$ has a statistically significant effect across all models and a high regression coefficient. It has both a positive and negative impact on the normalised rank depending on the environment. For standard tournaments, Figure 1.8 gives the distributions of several features for the winners of standard tournaments. The $C_r/C_{\text{mean}}$ distribution of the winner is also given in Figure 1.8. A value of $C_r/C_{\text{mean}} = 1$ implies that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament, and in standard tournaments, the median is 1. Therefore, an effective strategy in standard tournaments was the mean cooperator of its respective tournament.

The distributions of SSE and $CC$ to $D$ rate for the winners of standard tournaments are also given in Figure 1.8. The SSE distributions for the winners indicate that the strategy behaved in a ZD way in several tournaments, however, not constantly. The winners participated in matches where they did not try to extortionate their opponents. Furthermore, the $CC$ to $D$ distribution indicates that if a strategy were to defect against the winners they would reciprocate on average with a probability of 0.5.

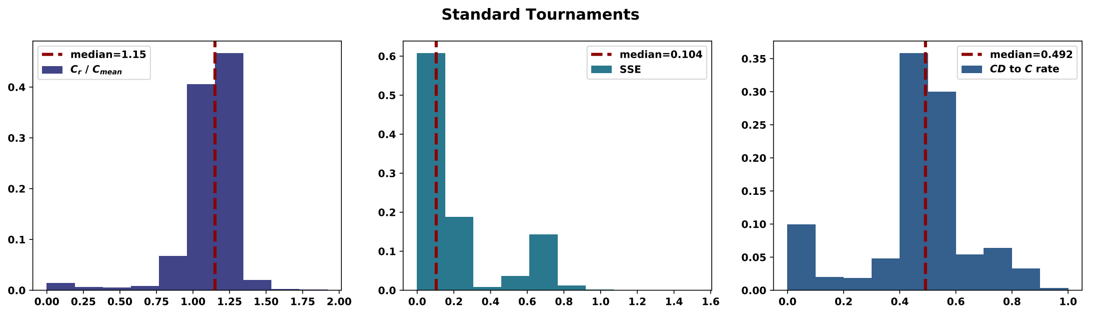



Figure 1.8: Distributions of $C_r/C_{\text{mean}}$, SSE and $CD$ to $C$ ratio for the winners of standard tournaments. A value of $C_r/C_{\text{mean}} = 1$ imply that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament. An SSE distribution skewed towards 0 indicates a extortionate behaviour by the strategy.

Similarly for the rest of the different tournaments types, and the entire data set the distributions of $C_r/C_{\text{mean}}$, SSE and $CD$ to $C$ ratio are given by Figures 1.9, 1.11, 1.12 and 1.13.

Based on the $C_r/C_{\text{mean}}$ distributions the successful strategies have adapted differently to the mean cooperator depending on the tournament type. In noisy tournaments where the median of the distribution is at 0.67, and thereupon the winners cooperated 67% of the time the mean cooperator did. In tournaments with noise and a probabilistic ending the winners cooperated 60%, whereas in settings that the type of the tournament can vary between all the types the winners cooperated 67% of the time the mean cooperator did. Lastly, in probabilistic ending tournaments above more defecting strategies prevail (Section 1.4), and this result is reflected here.

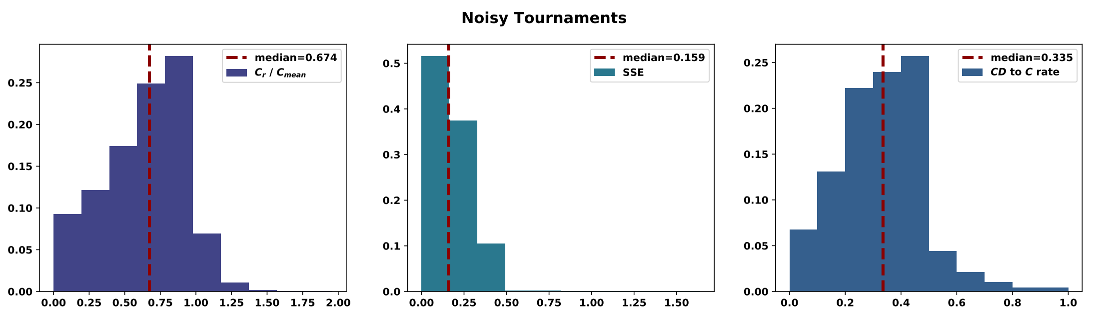
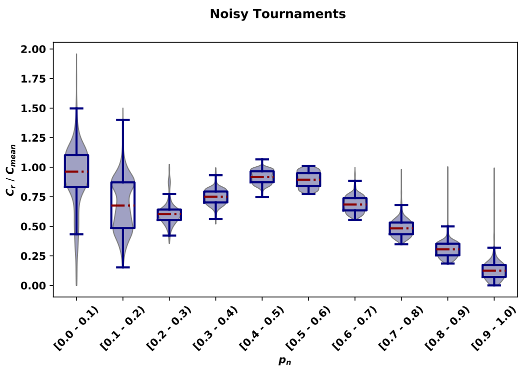


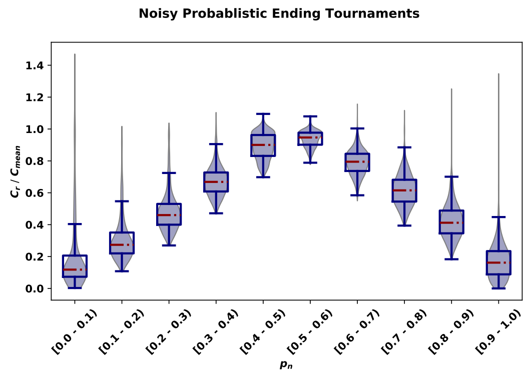

Figure 1.9: Distributions of $C_r/C_{\text{mean}}$, SSE and $CD$ to $C$ ratio for the winners of noisy tournaments.

The probability of noise has been observed to excessively affect optimal behaviour. In environments with considerable values of noise no strategy from our collection managed to perform sufficiently. Figure 1.10 gives the ratio $C_r/C_{\text{mean}}$ for the winners in tournaments with noise, over the probability of noise. From Figure 1.10a it is clear that the cooperating only 67% of the time the mean cooperator did is optimal only when $p_n \in [0.2, 0.4)$ and $p_n \in [0.6, 0.7]$. In environments with $p_n < 0.1$ the winners want to be close to the mean cooperator, similarly to standard tournaments, and as the probability of noise is exceeding 0.5 (the game becomes unreasonable) strategies should aim to be less and less cooperative.

Figure 1.10 gives $C_r/C_{\text{mean}}$ for the winners over $p_n$ in tournaments with noise and a probabilistic ending. The optimal proportions of cooperations are different now that the number of turns is not fixed, successful strategies want to be more defecting that the mean cooperator, that only changes when $p_n$ approaches 0.5. Figure 1.10 demonstrates how the adjustments to $C_r/C_{\text{mean}}$ change over the noise in the to the environment, and thus supports how important adapting to the environment is for a strategy to be successful.




(a) $C_r/C_{\text{mean}}$ distribution for winners in noisy tournaments over $p_n$.




(b) $C_r/C_{\text{mean}}$ distribution for winners in noisy probabilistic ending tournaments over $p_n$.

Figure 1.10: $C_r/C_{\text{mean}}$ distributions over intervals of $p_n$. These distributions model the optimal proportion of cooperation compared to $C_{\text{mean}}$ as a function of $(p_n)$.

The distributions of the SSE across the tournament types suggest that successful strategies exhibit some extortionate behaviour, but not constantly. ZDs are a set of strategies that are envious as they try to exploit their opponents. The winners of the tournaments considered in this work are envious, but not as much as many ZDs. This highlights why TFT's early tournament success fails to generalize – it never attempts to defect against a cooperating or exploitable opponent (e.g. Alternator). Moreover, many of the strategies in the library will not tolerate exploitation attempts. A clever strategy can achieve mutual cooperation with stronger strategies while also being able to exploit weaker strategies. This is why ZDs fail to appear in the top ranks – they try to exploit all opponents and cannot actively adapt back to mutual cooperation against stronger strategies, which requires more depth of memory. [3]

The distributions of the $CD$ to $C$ rate evaluate the behaviour of a successful strategy after its opponent has defected against it. In standard tournaments it was observed that a successful strategy reciprocates with a probability of 0.5. This is distinct between the tournament types. In tournaments with noise a strategy is less likely to cooperate following a defection compared to standard tournaments, and in probabilistic ending tournaments a strategy will reciprocate a defection. In a setting that the type of the tournament can vary between all the examined

[3] Note that ZDs also tend to perform poorly in population games for a similar reason: they attempt to exploit other players using ZDs, failing to form a cooperative subpopulation. This makes them good invaders but poor resisters of invasion.

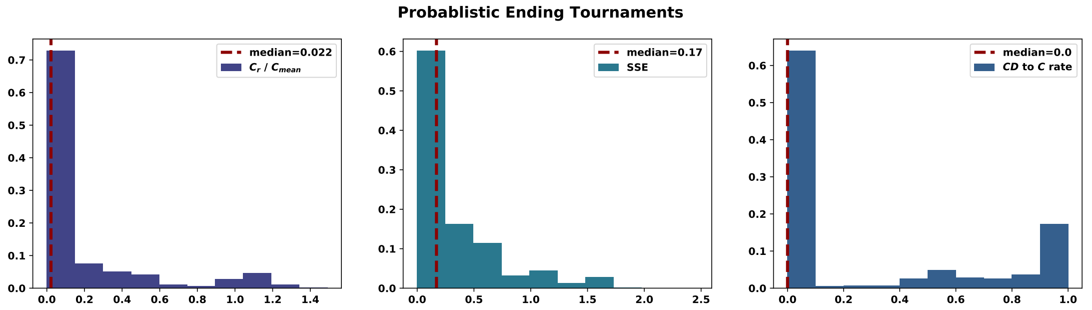



Figure 1.11: Distributions of $C_r/C_{\text{mean}}$, SSE and $CD$ to $C$ ratio for the winners of probabilistic ending tournaments.

types a winning strategy would reciprocate on average with a probability of 0.58.

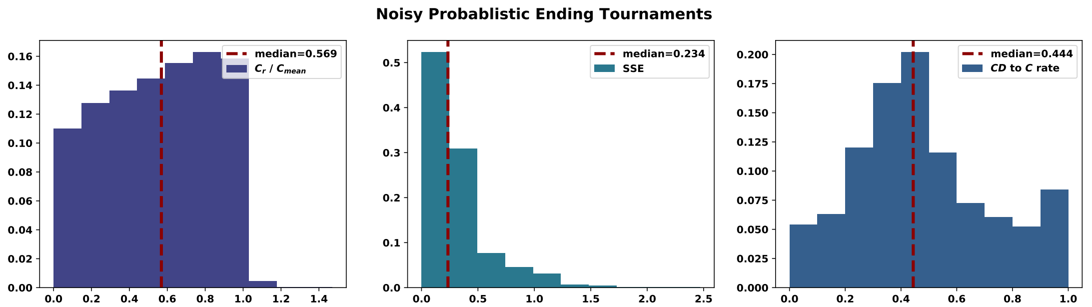



Figure 1.12: Distributions of $C_r/C_{\text{mean}}$, SSE and $CD$ to $C$ ratio for the winners of noisy probabilistic ending tournaments.

Further statistically significant features with strong effects include $C_r/C_{\text{min}}$, $C_r/C_{\text{max}}$, $C_{\text{min}}$ and $C_{\text{max}}$. These add more emphasis on how important it is for a a strategy to adapt to its environment. Finally, the features number of turns, repetitions and the probabilities of noise and the game ending had no significant effects based on the multivariate regression models.

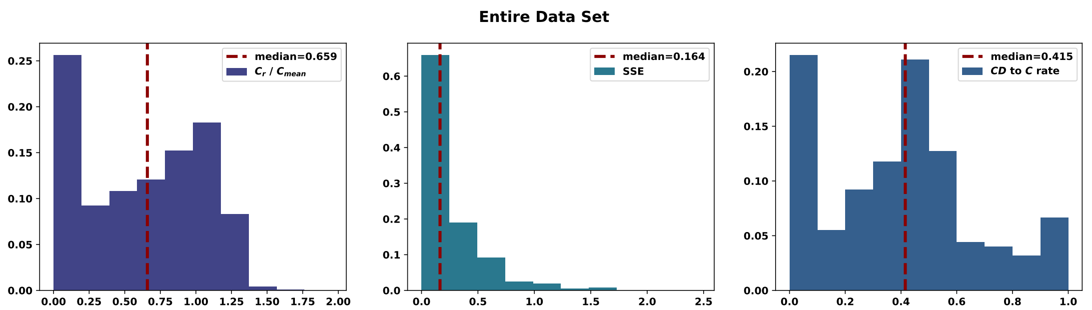



Figure 1.13: Distributions of $C_r/C_{\text{mean}}$, SSE and $CD$ to $C$ ratio for the winners over the tournaments of the entire data set.

A third method that evaluates the importance of the features in Table 1.5 using clustering and random forests can be found in the Appendix **??**. The results uphold the outcomes of the correlation and multivariate regression. It also evaluates the effects of the classifiers stochastic, make use of game, and make use of length which have not been evaluated by the methods above because there are binary variables. The results imply that they have no significant effect on a strategy's performance.

## 1.6 Chapter Summary

This manuscript has explored the performance of 195 strategies of the Iterated Prisoner's Dilemma in a large number of computer tournaments. We analyzed and extracted the salient features of the best performing strategies across various tournament types, casting the results in terms of Axelrod's original suggested features of good IPD strategies. Moreover, our results shed light on the historic performance of TFT, zero-determinant strategies, and memory one strategies generally. Strategies need to match their play to the cooperativeness of the tournament population and do so in a way that prevents or minimizes exploitation. Overall we see that complex or clever strategies can be effective, whether trained against a corpus of possible opponents or purposely designed to mitigate the impact of noise such as in the strategy DBS. Further, we showed that while the type of exploitation attempted by ZDs is not typically effective in tournaments, more sophisticated strategies capable of selectively exploiting weaker opponents while mutually cooperating with stronger opponents can be highly successful. This fact was also indicated numerically by the importance of the strategy feature SSE in the analysis of strategy features. These results highlight a central idea in evolutionary game theory in this context: the fitness landscape is a function of the population (where fitness in this case is tournament performance). While that may seem obvious now, it shows why historical tournament results on small or arbitrary populations of strategies have so often failed to produce generalizable results.

Highly noisy or tournaments with short matches favor less cooperative strategies. These environments mitigate the value of being nice. Uncertainty enables exploitation, reducing the ability of maintaining or enforcing mutual cooperation, while triggering grudging strategies to switch from typically cooperating to typically defecting. Accordingly, we find that in noisy tournaments the best performing players cooperate a lower rate than the tournament population on average. Nevertheless we found some strategies designed or trained for noisy environments were also highly ranked in noise-free tournaments. This indicates that strategy complexity is not necessarily a liability, rather it can confer adaptability to a more diverse set of environments.

In Section 1.4, the tournaments results were used to present the top performances. The data set contained results from four different settings, and these were also studied individually. In standard tournaments complex strategies trained using reinforcement learning ranked in the top spots. Some of these strategies ranked again in the top spots in probabilistic ending tournaments when a $p_e$ of less 0.1 was considered and in noisy tournaments when $p_n$ was less than 0.1. In probabilistic ending tournaments $p_e$ was designed to vary between 0 and 1. It was demonstrated that for values larger than 0.1, as stated in the Folk Theorem, defecting strategies were winning the tournaments because there was a high likelihood of the game ending in the next turn. In tournaments with noise the median ranks of the top 15 strategies had the highest values and the $r$ distributions were bimodal. The top rank strategies were performing both well and bad, and this indicates that in noisy tournaments where the noise can vary substantially, there were no strategies that can guarantee winning across a range of noise. However, if the probability of noise was constrained at 0.1 then strategies designed for noisy tournaments indeed performed well.

So what is the best way of playing the IPD? And is there a single dominant strategy for the

IPD? There was not a single strategy within the collection of the 195 strategies, that has managed to perform well in all the tournaments variations it competed in. Even if on average a strategy ranked highly in a specific environment it did not guarantee its success over the different tournament types. However, the results of sections 1.4 and 1.5 have demonstrated that there are properties associated with the success of strategies. A few of the properties that have been identified by this manuscript's analysis contradict the properties of Axelrod [14]. Namely, in Section 1.4 it was shown that trained strategies and strategies that decided their actions based on pre-designed strategies to maximise their utility dominated several tournaments across tournament types, hinting that successful IPD strategies are often clever or more complex than simple strategies like TFT. Most of the successful strategies highlighted in Section 1.4 were strategies that begin with cooperation.

Furthermore, in Section 1.4 and 1.5 it was shown that envious strategies performed well. Though these were not the most envious strategies in the tournaments (ZDs were included), these strategies benefited by being a bit envious. From Section 1.5 it was concluded that there is a significant importance in adapting to the environment, and more specifically in this work, to the mean cooperator. This section also demonstrated that a strategy should reciprocate, as suggested by Axelrod, but in some environments, such as standard and noisy, it should relax its readiness to do so.

Thus, the five properties successful strategies need to have in a IPD competition are: be nice, be provocable and contrite, be a little envious, be clever, and adapt to the environment (including the population of strategies).

The data set described in this work contains the largest number of IPD tournaments, to the authors knowledge, and it available at [26]. Further data mining could be applied and provide new insights in the field.

# Bibliography


[1] Lifl (1998) prison. `http://www.lifl.fr/IPD/ipd.frame.html`. Accessed: 2017-10-23.

[2] The prisoner's dilemma. http://www.prisoners-dilemma.com/, 2017.

[3] The Axelrod project developers . Axelrod: 4.4.0, April 2016.

[4] Eckhart A. Coopsim v0.9.9 beta 6. `https://github.com/jecki/CoopSim/`, 2015.

[5] M. Aberdour. Achieving quality in open-source software. *IEEE software*, 24(1):58–64, 2007.

[6] C. Adami and A. Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4:2193, 2013.

[7] D. Ashlock, J. A. Brown, and P. Hingston. Multiple opponent optimization of prisoner's dilemma playing agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):53–65, 2015.

[8] D. Ashlock and E. Y. Kim. Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies. *IEEE Transactions on Evolutionary Computation*, 12(5):647–659, Oct 2008.

[9] W. Ashlock and D. Ashlock. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. In *2006 IEEE International Conference on Evolutionary Computation*, pages 297–304. IEEE, 2006.

[10] W. Ashlock, J. Tsang, and D. Ashlock. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 135–142. IEEE, 2014.

[11] T. C. Au and D. Nau. Accident or intention: that is the question (in the noisy iterated prisoner's dilemma). In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 561–568. ACM, 2006.

[12] R. Axelrod. Effective choice in the prisoner's dilemma. *The Journal of Conflict Resolution*, 24(1):3–25, 1980.

[13] R. Axelrod. More effective choice in the prisoner's dilemma. *The Journal of Conflict Resolution*, 24(3):379–403, 1980.

[14] R. Axelrod. The emergence of cooperation among egoists. *American political science review*, 75(2):306–318, 1981.

[15] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. *Genetic Algorithms and Simulated Annealing*, pages 32–41, 1987.

[16] J. S. Banks and R. K. Sundaram. Repeated games, finite automata, and complexity. *Games and Economic Behavior*, 2(2):97–117, 1990.

[17] B. Beaufils, J. P. Delahaye, and P. Mathieu. Our meeting with gradual: A good strategy for the iterated prisoner's dilemma. 1997.

[18] J. Bendor, R. M. Kramer, and S. Stout. When in doubt... cooperation in a noisy prisoner's dilemma. *The Journal of Conflict Resolution*, 35(4):691–719, 1991.

[19] F. Benureau and N. P. Rougier. Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. *Frontiers in neuroinformatics*, 11:69, 2018.

[20] A. Carvalho, H. P. Rocha, F. T. Amaral, and F. G. Guimaraes. Iterated prisoner's dilemma-an extended analysis. 2013.

[21] C. Donninger. *Is it Always Efficient to be Nice? A Computer Simulation of Axelrod's Computer Tournament*. Physica-Verlag HD, Heidelberg, 1986.

[22] Merrill M. Flood. Some experimental games. *Management Science*, 5(1):5–26, 1958.

[23] M. R. Frean. The prisoner's dilemma without synchrony. *Proceedings of the Royal Society of London B: Biological Sciences*, 257(1348):75–79, 1994.

[24] D. Fudenberg and E. Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A Long-Run Collaboration On Long-Run Games*, pages 209–230. World Scientific, 2009.

[25] M. Gaudesi, E. Piccolo, G. Squillero, and A. Tonda. Exploiting evolutionary modeling to prevail in iterated prisoner's dilemma tournaments. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):288–300, 2016.

[26] N. E. Glynatsi. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results. `https://doi.org/10.5281/zenodo.3516652`, October 2019.

[27] Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E. Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PLOS ONE*, 12(12):1–33, 12 2017.

[28] C. Hilbe, M. A. Nowak, and A. Traulsen. Adaptive dynamics of extortion and compliance. *PLOS ONE*, 8(11):1–9, 11 2013.

[29] G. Kendall, X. Yao, and S. Y. Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[30] V. A. Knight, M. Harper, N. E. Glynatsi, and J. Gillard. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR*, abs/1904.00973, 2019.

[31] D. Kraines and V. Kraines. Pavlov and the prisoner's dilemma. *Theory and decision*, 26(1):47–79, 1989.

[32] S. Kuhn. Prisoner's dilemma. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

[33] J. Li, P. Hingston, S. Member, and G. Kendall. Engineering Design of Strategies for Winning Iterated Prisoner ' s Dilemma Competitions. 3(4):348–360, 2011.

[34] J. Li and G. Kendall. A strategy with novel evolutionary features for the iterated prisoner's dilemma. *Evolutionary Computation*, 17(2):257–274, 2009.

[35] P. Mathieu and J. P. Delahaye. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation*, 20(4):12, 2017.

[36] J. H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1):87 – 112, 1996.

[37] S. Mittal and K. Deb. Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 13(3):554–565, 2009.

[38] P. Molander. The optimal level of generosity in a selfish, uncertain environment. *The Journal of Conflict Resolution*, 29(4):611–618, 1985.

[39] J. H. Nachbar. Evolution in the finitely repeated prisoner's dilemma. *Journal of Economic Behavior & Organization*, 19(3):307–326, 1992.

[40] M. Nowak and K. Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58, 1993.

[41] M. A. Nowak and K. Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355:250–253, January 1992.

[42] prase. Prisoner's dilemma tournament results. `https://www.lesswrong.com/posts/hamma4XgeNrsvAJv5/prisoner-s-dilemma-tournament-results`, 2011.

[43] W. H. Press and F. G Dyson. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.

[44] A. J. Robson. Efficiency in evolutionary games: Darwin, nash and the secret handshake. *Journal of theoretical Biology*, 144(3):379–396, 1990.

[45] R. Selten and P. Hammerstein. Gaps in harley's argument on evolutionarily stable learning rules and in the logic of "tit for tat". *Behavioral and Brain Sciences*, 7(1):115–116, 1984.

[46] D. W. Stephens, C. M. McLinn, and J. R. Stevens. Discounting and reciprocity in an iterated prisoner's dilemma. *Science*, 298(5601):2216–2218, 2002.

[47] A. J. Stewart and J. B. Plotkin. Extortion and cooperation in the prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(26):10134–10135, 2012.

[48] E. Tzafestas. Toward adaptive cooperative behavior. 2:334–340, Sep 2000.

[49] E. Tzafestas. Toward adaptive cooperative behavior. *From Animals to animals: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior (SAB-2000)*, 2:334–340, 2000.

[50] P. Van-Den-Berg and F. J. Weissing. The importance of mechanisms for the evolution of cooperation. In *Proc. R. Soc. B*, volume 282, page 20151382. The Royal Society, 2015.

# Appendix A

# List of Strategies

## A.1 List of strategies considered in Chapter 1

The strategies considered in Chapter 1, which are from APL version 3.0.0.

1. $\phi$ [3]
2. $\pi$ [3]
3. $e$ [3]
4. ALLCorALLD [3]
5. Adaptive [33]
6. Adaptive Pavlov 2006 [29]
7. Adaptive Pavlov 2011 [33]
8. Adaptive Tit For Tat: 0.5 [49]
9. Aggravater [3]
10. Alexei [42]
11. Alternator [14, 37]
12. Alternator Hunter [3]
13. Anti Tit For Tat [28]
14. AntiCycler [3]
15. Appeaser [3]
16. Arrogant QLearner [3]
17. Average Copier [3]
18. Backstabber [3]
19. Better and Better [1]
20. Bully [39]
21. Calculator [1]
22. Cautious QLearner [3]
23. Champion [13]
24. CollectiveStrategy [34]
25. Contrite Tit For Tat [?]
26. Cooperator [14, 37, 43]
27. Cooperator Hunter [3]
28. Cycle Hunter [3]
29. Cycler CCCCCD [3]
30. Cycler CCCD [3]
31. Cycler CCCDCD [3]
32. Cycler CCD [37]
33. Cycler DC [3]
34. Cycler DDC [37]
35. DBS [11]
36. Davis [12]
37. Defector [14, 37, 43]
38. Defector Hunter [3]
39. Double Crosser [3]
40. Desperate [50]
41. DoubleResurrection [4]
42. Doubler [1]
43. Dynamic Two Tits For Tat [3]
44. EasyGo [33, 1]
45. Eatherley [13]
46. Eventual Cycle Hunter [3]
47. Evolved ANN [3]
48. Evolved ANN 5 [3]
49. Evolved ANN 5 Noise 05 [3]
50. Evolved FSM 16 [3]
51. Evolved FSM 16 Noise 05 [3]
52. Evolved FSM 4 [3]
53. Evolved HMM 5 [3]
54. EvolvedLookerUp1 1 1 [3]

55. EvolvedLookerUp2 2 2 [3]
56. Eugine Nier [42]
57. Feld [12]
58. Firm But Fair [23]
59. Fool Me Forever [3]
60. Fool Me Once [3]
61. Forgetful Fool Me Once [3]
62. Forgetful Grudger [3]
63. Forgiver [3]
64. Forgiving Tit For Tat [3]
65. Fortress3 [9]
66. Fortress4 [9]
67. GTFT [25, 40]
68. General Soft Grudger [3]
69. Gradual [17]
70. Gradual Killer [1]
71. Grofman[12]
72. Grudger [12, 16, 17, 50, 33]
73. GrudgerAlternator [1]
74. Grumpy [3]
75. Handshake [44]
76. Hard Go By Majority [37]
77. Hard Go By Majority: 10 [3]
78. Hard Go By Majority: 20 [3]
79. Hard Go By Majority: 40 [3]
80. Hard Go By Majority: 5 [3]
81. Hard Prober [1]
82. Hard Tit For 2 Tats [47]
83. Hard Tit For Tat [2]
84. Hesitant QLearner[3]
85. Hopeless [50]
86. Inverse [3]
87. Inverse Punisher [3]
88. Joss [12, 47]
89. Knowledgeable Worse and Worse [3]
90. Level Punisher [4]
91. Limited Retaliate 2 [3]
92. Limited Retaliate 3 [3]
93. Limited Retaliate [3]
94. MEM2 [?]
95. Math Constant Hunter [3]
96. Meta Hunter Aggressive [3]
97. Meta Hunter [3]
98. Meta Majority [3]
99. Meta Majority Finite Memory [3]
100. Meta Majority Long Memory [3]
101. Meta Majority Memory One [3]
102. Meta Minority [3]
103. Meta Mixer [3]
104. Meta Winner [3]
105. Meta Winner Deterministic [3]
106. Meta Winner Ensemble [3]
107. Meta Winner Finite Memory [3]
108. Meta Winner Long Memory [3]
109. Meta Winner Memory One [3]
110. Meta Winner Stochastic [3]
111. NMWE Deterministic [3]
112. NMWE Finite Memory [3]
113. NMWE Long Memory [3]
114. NMWE Memory One [3]
115. NMWE Stochastic [3]
116. Naive Prober [33]
117. Negation [2]
118. Nice Average Copier [3]
119. Nice Meta Winner [3]
120. Nice Meta Winner Ensemble [3]
121. Nydegger [12]
122. Omega TFT [29]
123. Once Bitten [3]
124. Opposite Grudger [3]
125. PSO Gambler 1 1 1 [3]
126. PSO Gambler 2 2 2 [3]
127. PSO Gambler 2 2 2 Noise 05 [3]
128. PSO Gambler Mem1 [3]
129. Predator [9]
130. Prober [33]
131. Prober 2 [1]
132. Prober 3 [1]
133. Prober 4 [1]
134. Pun1 [9]

135. Punisher [3]
136. Raider [10]
137. Random Hunter [3]
138. Random: 0.5 [12, 49]
139. Remorseful Prober [33]
140. Resurrection [4]
141. Retaliate 2 [3]
142. Retaliate 3 [3]
143. Retaliate [3]
144. Revised Downing [12]
145. Ripoff [8]
146. Risky QLearner [3]
147. SelfSteem [20]
148. ShortMem [20]
149. Shubik [12]
150. Slow Tit For Two Tats [3]
151. Slow Tit For Two Tats 2 [1]
152. Sneaky Tit For Tat [3]
153. Soft Go By Majority [14, 37]
154. Soft Go By Majority 10 [3]
155. Soft Go By Majority 20 [3]
156. Soft Go By Majority 40 [3]
157. Soft Go By Majority 5 [3]
158. Soft Grudger [33]
159. Soft Joss [1]
160. SolutionB1 [7]
161. SolutionB5 [7]
162. Spiteful Tit For Tat [1]
163. Stalker [**?**]
164. Stein and Rapoport [12]
165. Stochastic Cooperator [6]
166. Stochastic WSLS [3]
167. Suspicious Tit For Tat [17, 28]
168. TF1 [3]
169. TF2 [3]
170. TF3 [3]
171. Tester [13]
172. ThueMorse [3]
173. ThueMorseInverse [3]
174. Thumper [8]
175. Tit For 2 Tats (**Tf2T**) [14]
176. Tit For Tat (**TfT**) [12]
177. Tricky Cooperator [3]
178. Tricky Defector [3]
179. Tullock [12]
180. Two Tits For Tat (**2TfT**) [14]
181. VeryBad [20]
182. Willing [50]
183. Win-Shift Lose-Stay (**WShLSt**) [33]
184. Win-Stay Lose-Shift (**WSLS**) [31, 40, 47]
185. Winner12 [35]
186. Winner21 [35]
187. Worse and Worse[1]
188. Worse and Worse 2[1]
189. Worse and Worse 3[1]
190. ZD-Extort-2 v2 [32]
191. ZD-Extort-2 [47]
192. ZD-Extort-4 [3]
193. ZD-GEN-2 [32]
194. ZD-GTFT-2 [47]
195. ZD-SET-2 [32]

# Appendix B

# Correlation coefficients of features in Chapter 1

A graphical representation of the correlation coefficients for the features of Table **??**.

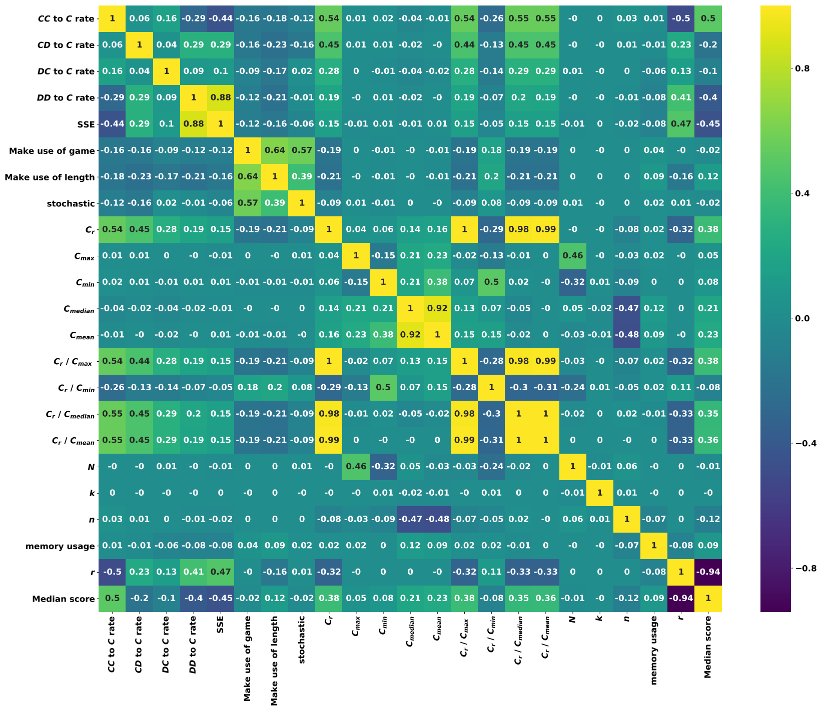



Figure B.1: Correlation coefficients of measures in Table **??** for standard tournaments

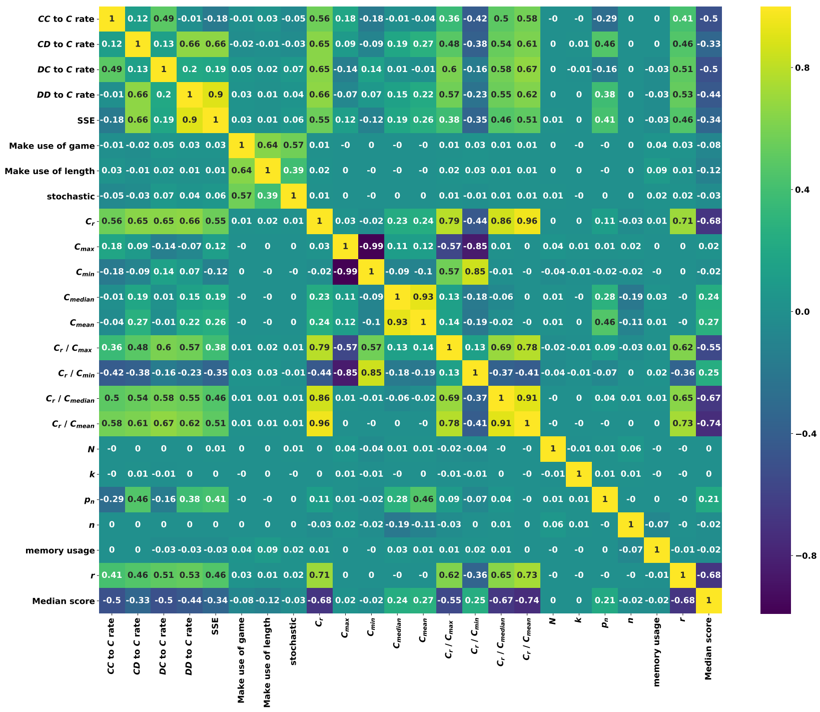



Figure B.2: Correlation coefficients of measures in Table ?? for noisy tournaments

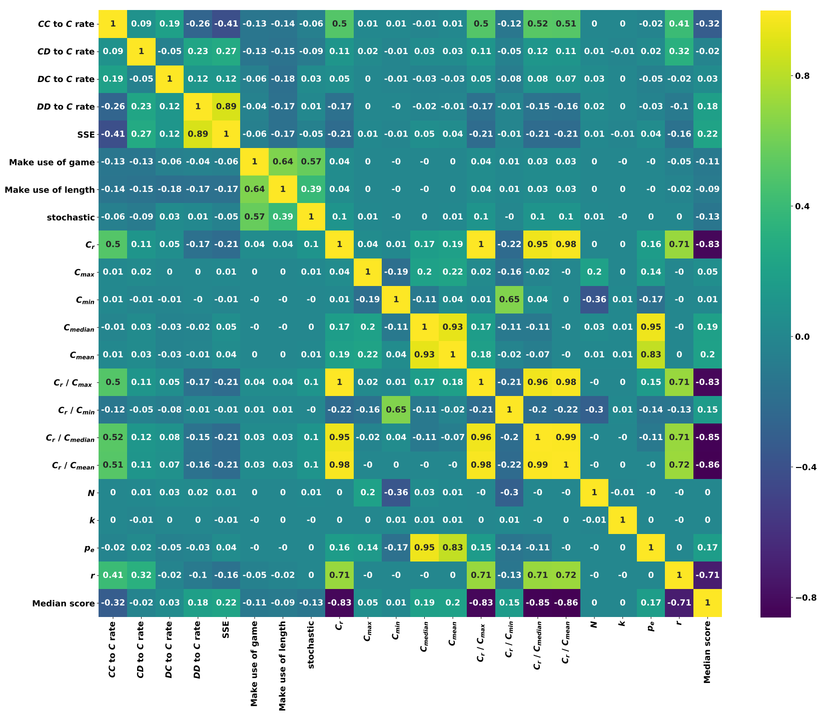



Figure B.3: Correlation coefficients of measures in Table ?? for probabilistic ending tournaments

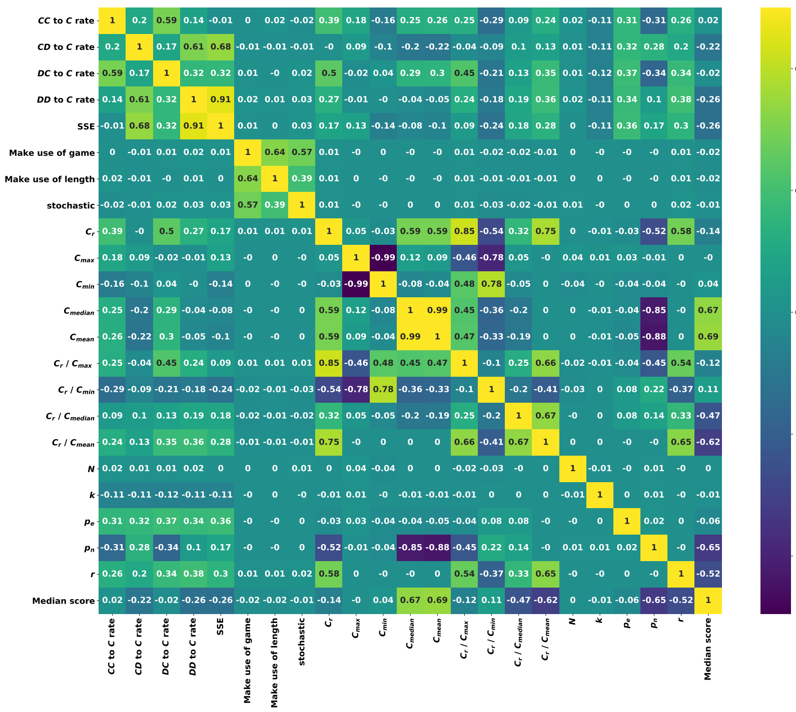



Figure B.4: Correlation coefficients of measures in Table **??** for noisy probabilistic ending tournaments

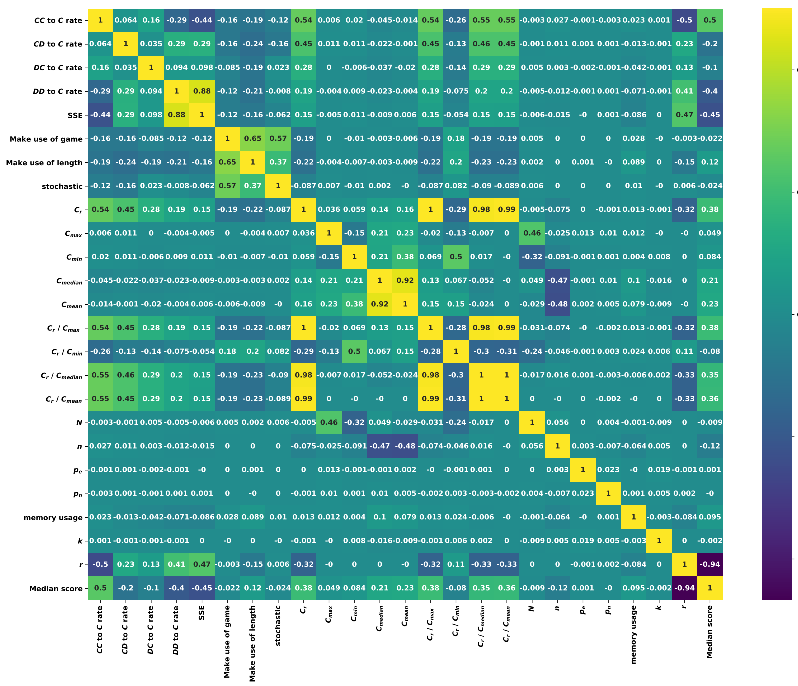



Figure B.5: Correlation coefficients of measures in Table **??** for data set