# Morality Metrics On Iterated Prisoner's Dilemma Players

Tyler Singer-Clark

June 5, 2014

## Abstract

Research regarding Iterated Prisoner's Dilemma (IPD) tournaments generally focuses on the objective performance of strategies. This paper instead discusses moral judgement of strategies, and analyzes IPD stratagies' behaviors through clearly defined morality functions. No single moral code is accepted by everyone, so multiple moral views are represented in this paper. Its purpose is not to argue for a specific moral view (or to attempt to cover all possible views), but rather to present a useful method of moral analysis as well as some interesting, mathematically well-defined morality functions corresponding (or at least relevant) to real-world views represented in society.

## 1 Introduction

A *Prisoner's Dilemma* (PD) is a model of an encounter between two entities (for example, human beings) that allows each player to simultaneously choose between two options: $C$ (Cooperate) or D (Defect). The objective payoffs for the combinations of choices are $T$ (Temptation), $R$ (Reward), $P$ (Punishment), and $S$ (Sucker), which obey the constraint $T > R > P > S$. For mutual defection, each player receives $P$. For mutual cooperation, each player receives $R$. If Player 1 defects and Player 2 cooperates, Player 1 receives $T$ while Player 2 receives $S$. An *Iterated Prisoner's Dilemma* (IPD) is when two players repeatedly interact with one another, remembering the history of their interaction so far, so they can each base their decisions on past moves. The additional constraint $T + S < 2R$ ensures that both players repeatedly choosing to cooperate is more beneficial to each than them taking turns taking advantage of one another. An *IPD tournament* takes several programmed strategies (bots) and allows them to interact with one another, each attempting to maximize their own total score. For reference, see Robert Axelrod's 1984 book *The Evolution Of Cooperation*, which includes a helpful introduction to IPD as well as the framework for the IPD tournaments discussed in this paper.[1]
By simply running such tournaments and observing the scores and rankings, one can rate the objective success of given strategies in various environments of other bots. But given that an IPD can model human interactions, it seems natural to ask not only about the objective success of each bot, but also about the ethics of each bot's behavior.[1] IPD gives us a simple model of human behavior that is easier to analyze than many real-world situations. A software system was built that is made up of a virtual arena in which to host IPD tournaments, a collection of implemented bots (and the ability to easily create more), and a morality calculation tool through which the results of a tournament can be sent in order to see a set of morality functions evaluated on each bot.[2] This system makes easy the process of running tournaments with any desired environment of bots, and morally judging each bot in accordance with various belief systems.

[1] This paper deals with "utilitarian ethics," in that the morality functions look only at behavior, and not at the code behind the bots' choices. Morality in this sense is a function of behavior and behavior depends on the other bots in the environment, so this type of morality is not simply an attribute of a strategy but is actually relative to an environment of other bots.

[2] It is assumed that the bots being judged, as well as their creators, are unaware of any omniscent observer or any "score" besides the objective, and thus their objective is unaltered. One might conclude that this weakens the analogy between IPD players and human beings because humans can acknowledge morality and act accordingly. But because the objective payoffs are not bound to any specific meaning, they could simply be defined to capture the effects of actions on players' feelings due to any moral code they strive to obey, and thus the analogy is not weakened by this assumption.

Some people believe kindness is unconditionally the correct moral choice (a "Jesus"-like point of view). A slightly less strict rule is that the amount of kindness required is only as much as that received from the target in question. This allows retaliation, but does not require it. Still others consider justice a necessary part of morality (as "Moses" might maintain). This school of thought, perhaps stemming from the idea that soft treatment of offenders encourages them to continue offending, would dictate that harshness be reciprocated. This paper presents well-defined morality functions that express such moral views and were implemented in the above-mentioned morality calculation tool. The results of both conducting an IPD tournament using said system and performing morality calculations on this tournament's output are then described and analyzed. The experiment, including the list of participating bots and the set of morality functions, is not exhaustive, but can hopefully be used as a framework for a new type of ethical analysis which uses IPD tournaments and morality functions to model human behavior and moral judgement.

# 2 Meet The Bots

The bots listed below represent a wide range of sophistication and design, and are gathered from various sources.

## 2.1 ALL_D and ALL_C

ALL_D unconditionally defects every turn. This is the least cooperative strategy possible, and it rarely receives anything but the Punishment after the first few turns.

ALL_C unconditionally cooperates every turn. This is the most cooperative strategy possible, and while ALL_C elicits cooperation from kind partners, it is easy to take advantage of ALL_C.

Because these bots do not take into account their partner's moves, the optimal strategy toward them is to defect every turn.

## 2.2 RANDOM

RANDOM takes an initialization parameter *p_cooperate* and each turn cooperates with probability p_cooperate, independent of its partner's moves. RANDOM acts independently of its partner's moves, so the optimal strategy toward it is to defect every turn.

## 2.3 PAVLOV

PAVLOV follows the mantra of "win-stay, lose-shift",[3] which means it repeats its most recent action after receiving a good outcome (Temptation or Reward) and changes its action after a bad outcome (Punishment or Sucker).[3] Another way to view this behavior is as follows: PAVLOV cooperates after it and its partner's previous moves match, and defects after it and its previous moves differ. This means it will cooperate after mutual defection, leaving it vulnerable to strategies that often defect consecutively. PAVLOV also defects again after taking advantage of its partner, so if it successfully exploits a bot, it continues trying to push its luck. PAVLOV defaults to cooperation on the first move.

## 2.4 TIT_FOR_TAT (and Variations)

TIT_FOR_TAT is an extremely simple strategy that showed great performance and robustness in Axelrod's original tournaments.[1] TIT_FOR_TAT cooperates on its first move and then simply returns its partner's previous move. TIT_FOR_TAT's score in an interaction is never higher than that of its partner. TIT_FOR_TAT also protects itself from being taken advantage of more than one more time than TIT_FOR_TAT is able to take advantage of its partner.

Many variations on TIT_FOR_TAT exist,[2] such as TIT_FOR_TWO_TATS and TWO_TITS_FOR_TAT. TIT_FOR_TWO_TATS only defects after its partner defects twice in a row. It is more forgiving than TIT_FOR_TAT but can be hurt by a bot that alternates cooperation and defection. TWO_TITS_FOR_TAT responds to every defection by its partner with two defections of its own. This extra retaliation can cause it to miss out on potential mutual cooperation.

SUSPICIOUS_TIT_FOR_TAT is the same as TIT_FOR_TAT except that it defaults to defection on the first turn. This slight change is actually extremely significant, and generally causes SUSPICIOUS_TIT_FOR_TAT to rank much lower than

[3] This resembles the classical conditioning phenomenon studied by Ivan Pavlov, giving PAVLOV its name.

TIT_FOR_TAT itself, because many bots do not want to cooperate with a bot that defects right off the bat.

GENEROUS_TIT_FOR_TAT differs from TIT_FOR_TAT in that, after its partner defects, GENEROUS_TIT_FOR_TAT cooperates with some probability *p_generous*, allowing it to salvage potentially rewarding interactions that TIT_FOR_TAT essentially gives up on. JOSS[4] is GENEROUS_TIT_FOR_TAT's evil twin. After its partner cooperates, JOSS defects with some probability *p_sneaky* to see what it can get away with. Like SUSPICIOUS_TIT_FOR_TAT, this seemingly small extra defection gets JOSS into unnecessary trouble with retaliatory partners.

## 2.5 MAJORITY

MAJORITY begins with cooperation, and from then on cooperates as long as its partner has cooperated more than they have defected. MAJORITY can be either "soft" or "hard", and this determines MAJORITY's behavior when its partner has cooperated and defected an equal number of times.

## 2.6 TESTER

TESTER was submitted to Axelrod's tournament by David Gladstein.[1] TESTER initially defects to see what its partner will do. If TESTER's partner ever defects, TESTER apologizes by cooperating and then mirrors its partner's moves thereafter. If the other player does not retaliate, TESTER cooperates twice but then alternates cooperation and defection from then on. Retaliatory partners can elicit cooperation from TESTER because it realizes that defection is not profitable, but the initial defection can still cause the interaction to never settle on mutual cooperation, leading TESTER to perform poorly in many environments.

## 2.7 FRIEDMAN

Also known as GRIM_TRIGGER, FRIEDMAN[5] defaults to cooperation until the first defection by its partner, after which FRIEDMAN defects unconditionally. While FRIEDMAN is "nice,"[6] it is maximally unforgiving, which stops it from ever being able to salvage an interaction once a defection occurs.

## 2.8 EATHERLY and CHAMPION

EATHERLY and CHAMPION scored very well in Axelrod's tournament,[1] and both follow a similar idea. EATHERLY keeps track of its partner's defection rate (the fraction of total turns its partner has defected) so that after its partner defects, EATHERLY can defect with probability equal to its partner's defection rate. CHAMPION does something similar, except it begins with a short period of unconditional cooperation, mirrors its partner's moves for another short period, and thereafter does the same as EATHERLY, except if its partner has cooperated more than 60% of the time then CHAMPION cooperates even after its partner defects.

# 3 Morality Metrics

## 3.1 Cooperation Rate

Overall cooperation rate is one of the simplest possible morality metrics, while still being one of the most informative. If $CR(b)$ is the cooperation rate of bot $b$,

$$CR(b) = \frac{C(b)}{TT}$$

where $C(b)$ is the total number of turns bot $b$ chose to cooperate throughout the whole tournament, and $TT$ is the total number of turns a single bot plays throughout the whole tournament.

ALL_C is trivially the highest scorer by this morality metric ($CR(ALL\_C) = 1.0$ guaranteed), but there are other more interesting bots that have very high cooperation rates without sacrificing performance the way ALL_C does.

A subtle point about this metric is that it does not take into account how the cooperations are distributed between partners. Consider bots EVEN and BIAS, where EVEN cooperates with $b_0$ and $b_1$ each 50 times while BIAS cooperates with $b_0$ 100

[4] The name comes from Johann Joss, who submitted this strategy to Axelrod's tournament.

[5] FRIEDMAN is the name Axelrod uses for this strategy.

[6] Axelrod uses the term "nice" to describe a bot who is never the first to defect in any given interaction.

times and with $b_1$ 0 times. Under the metric of cooperation rate, EVEN and BIAS are judged to be equal, but their behaviors are vastly different. BIAS did not even give $b_1$ a chance, and EVEN did not fully cooperate with either $b_0$ or $b_1$, and these features end up perfectly cancelling.

## 3.2 Good-Partner Rating

A bot's good-partner rating is the fraction of partnerships in which that bot cooperated at least as much as its partner.[7] Just as for cooperation rate, at any given turn cooperating instead of defecting cannot lower a bot's good-partner rating. But because good-partner rating takes into account how cooperations are distributed, higher cooperation rate does not necessarily imply higher good-partner rating. Good-partner rating is a quantitative representation of the idea that defections are justified when one's partner defects just as much. Analogously, a bot can have a perfect 1.0 good-partner rating without having a perfect 1.0 cooperation rate, as long as the defections performed by that bot occur only with partners who defect at least as much.

When situations involve defection, those attempting to cast judgement sometimes factor in who the instigator was, rather than look only at the total number of defections by each party. From this perspective, a flaw of good-partner rating as a morality metric is that it does not depend on the order of moves, and instead only compares totals. Consider bots FRIEDMAN and GENEROUS_JOSS. As described above, FRIEDMAN cooperates until its partner defects, after which FRIEDMAN always defects. GENEROUS_JOSS is a hybrid of GENEROUS_TIT_FOR_TAT and JOSS that begins with cooperation and thereafter mirrors its partner's previous move, except after its partner defects it cooperates with some probability p_generous and after its partner cooperates it defects with some probability p_sneaky. FRIEDMAN and GENEROUS_JOSS will begin by cooperating, but at some point GENEROUS_JOSS will throw in one of its sneaky defections, triggering FRIEDMAN's never-ending retaliation. Mutual defection will ensue, but GENEROUS_JOSS will throw in an occasional generous cooperation to no avail. Had GENEROUS_JOSS not thrown the first punch, the interaction would have been one of complete mutual cooperation, so it is fair to say that GENEROUS_JOSS was the instigator. But GENEROUS_JOSS will end up with more cooperations than FRIEDMAN because of its occasional attempts to rectify the situation, so the interaction will count toward GENEROUS_JOSS's good-partner rating and not FRIEDMAN's, even though it was GENEROUS_JOSS's fault the defections started in the first place. So even though good-partner rating tries to forgive bots for defecting against uncooperative partners, because it has no notion of "who started it" it does not capture the full view of many who believe in justified retaliation.

## 3.3 Eigenjesus Rating

Eigenjesus rating is a recursively defined morality metric that always favors cooperation, and gives more weight to cooperations with moral bots than to cooperations with immoral bots.[8] The moral view analogous to eigenjesus rating is that which maintains that kindness is always better, especially toward others who are themselves kind and thus more deserving of receiving kindness.

The mathematical idea behind eigenjesus rating is similar to that of Google PageRank.[4] In PageRank, pages distribute their "vote" for other pages by linking to them. In eigenjesus rating, bots distribute their "vote" to other bots who cooperated with them. The amount bot $b_i$ contributes to another bot $b_j$'s eigenjesus rating is proportional to $b_i$'s own eigenjesus rating[9] and to $b_j$'s cooperation rate when interacting with $b_i$.[10] Consider an IPD tournament of $n$ bots with cooperation matrix $C \in [0,1]^{n\times n}$, where $C = (c_{ij})$ and $c_{ij}$ is bot $b_i$'s cooperation rate when interacting with bot $b_j$. Denote $C$'s principal eigenvector as $\vec{v} = [v_1, \ldots, v_n]$.[11] Then $v_i$ is bot $b_i$'s

[7] The good-partner rating calculation excludes interactions between a bot with its own clone. This is because the probabilities that a bot cooperates more or less than its own clone are exactly equal, regardless of the strategy, so including such interactions adds nothing of value to the metric.

[8] The recursiveness of the definition is in the fact that how moral or immoral other bots are is judged from their own eigenjesus rating.

[9] This is analogous to the way a webpage with higher PageRank can contribute more to another page's PageRank.

[10] This is analogous to the way a webpage only contributes to the PageRank of pages to which it links, but in PageRank the linking is binary (1 for a link, 0 for no link) while clearly cooperation rate can be 1, 0, or anywhere in between.

[11] The eigenvalue corresponding to $\vec{v}$ can be used as a measure of the total goodness in the environment of an IPD tour-

eigenjesus rating.

Because bot $b_0$ cooperating more with bot $b_1$ causes $b_1$ to contribute more to $b_0$'s eigenjesus rating, it is clear that cooperating is always better for a bot's eigenjesus rating. Thus, ALL_C is again the strategy with the highest possible rating, independent of the environment. One way eigenjesus rating differs from pure cooperation rate as a morality metric is that it has the interesting property of caring less about how a bot behaves with very uncooperative bots. For example, ALL_D gets an eigenjesus rating of 0.0 because it never cooperates and thus gets a contribution of 0.0 from every other bot. This means that cooperating with ALL_D does nothing for another bot's eigenjesus rating. Essentially, ALL_D's lack of kindness means it has no say in who else is considered kind, so other bots are not judged any worse for defending themselves against ALL_D. Eigenjesus is more concerned with how bots act when partnered with cooperative bots. Eigenjesus especially looks down on exploitation of others. To see this, consider the fact that if a bot is being consistently exploited by one bot, it likely is also being exploited by or mutually cooperating with the other bots as well, giving it a high eigenjesus rating and thus making it a valuable contributor to other bots' eigenjesus ratings. So when one bot exploits another, it is lowering its cooperation rate with exactly the type of bot that would contribute most to its eigenjesus rating.

## 3.4 Eigenmoses Rating

Eigenmoses rating shares the same analogy to PageRank as eigenjesus rating does, but instead of using the principal eigenvector of the cooperation matrix $C$, it uses the principal eigenvector of the matrix $D = (d_{ij})$, where

$$d_{ij} = 2(c_{ij} - 0.5)$$

$D$ still represents how much each bot cooperated in each interaction, but while $C \in [0,1]^{n \times n}$, instead $D \in [-1,1]^{n \times n}$. Uncooperative behavior is now represented by a negative number, instead of a small positive number.

This slight mathematical modification has large moral consequences. Perhaps the most important consequence is that, unlike the three previously discussed morality metrics, eigenmoses rating does not always favor cooperation. Cooperations only add to a bot's eigenmoses rating if they are with a partner with positive eigenmoses rating themselves. In fact, cooperations with a bot whose eigenmoses rating is negative actually lower a bot's eigenmoses rating. So eigenmoses rating requires defection from IPD players when paired with certain partners, specifically those who often defect themselves. Eigenmoses demands justice.

This attitude reflects several real-world moral views. One is that evil simply deserves to be treated with evil, which is certainly a satisfying symmetry, and is likely the root of the common human desire for revenge in many cases. Another is the conditioning argument that if someone acts selfishly and it pays off for them because too many people continued to cooperate with them anyway (often referred to as "enabling"), they will learn that selfishness is beneficial and continue to display unkind habits. This reasoning about conditioning seems inappropriate for the IPD tournament model used in this paper, since the interactions are isolated to each bot pair. Because this isolation means that behavior can only be trained within interactions and cannot be trained across interactions, a given bot can only affect its own interaction with its partner, so no other bots will suffer due to this bot's enabling of its partner. This all changes when IPD is extended to an ecological or evolutionary model[1][3] in which the performance of a strategy in a tournament has an impact on its concentration in subsequent tournament environments. Less retaliatory bots (bots that would be criticized under eigenmoses rating for cooperating with defectors) make the environment profitable for selfish bots, so selfish strategies can prosper, and sometimes even take over, if the environment has too high a concentration of strategies soft on defection, and this ruins the environment for everyone, including the bots who were not involved in the enabling. This discussion highlights the fact that varying the particular IPD model used can yield significant changes in the notion of morality and justifications of certain metrics.

# 4 Results

The results in the tables below are from a tournament with payoffs $T = 5$, $R = 3$, $P = 1$, $S = 0$, and discount parameter $w = 0.995$. The number of

nament.

meetings was 5, and the interaction lengths were [417, 160, 37, 277, 108]. The bots that have a number or word appended to their name use that number or word as their initialization parameter. For example, RANDOM_0.5 has *p_cooperate* = 0.5.

## 4.1 Cooperation Rate

Table 2 shows the bots ordered by cooperation rate. First, there is a clear connection between higher cooperation rate and objective success in the tournament. Excluding dumb bots,[12] the bots with the nine highest cooperation rates are the nine top finishers in the tournament.[13] But, as would be expected, a high cooperation rate on its own is not enough to guarantee success, shown by the position of ALL_C (cooperation rate of 1.0) and RANDOM_0.8 (cooperation rate of 0.801) in the bottom half of the tournament.

| **Bot** | **Cooperation Rate** |
|---|---|
| ALL_C | 1.0 |
| EATHERLY | 0.893 |
| CHAMPION | 0.884 |
| GENEROUS_TIT_FOR_TAT_0.3 | 0.883 |
| MAJORITY_SOFT | 0.869 |
| TIT_FOR_TWO_TATS | 0.861 |
| GENEROUS_TIT_FOR_TAT_0.1 | 0.829 |
| RANDOM_0.8 | 0.801 |
| PAVLOV | 0.772 |
| MAJORITY_HARD | 0.770 |
| TIT_FOR_TAT | 0.756 |
| TWO_TITS_FOR_TAT | 0.651 |
| FRIEDMAN | 0.608 |
| TESTER | 0.551 |
| RANDOM_0.5 | 0.501 |
| SUSPICIOUS_TIT_FOR_TAT | 0.467 |
| JOSS_0.1 | 0.444 |
| JOSS_0.3 | 0.251 |
| RANDOM_0.2 | 0.198 |
| ALL_D | 0.0 |

Table 2: Bot List Sorted By Cooperation Rate

Strategies that did not necessarily defect following every one of their partner's defections, like GENEROUS_TIT_FOR_TAT_0.3 and EATHERLY, were the ones with the highest cooperation rates, again excluding dumb strategies like ALL_C. For some bots, the mechanism that lead them to have higher cooperation rates was the same feature that allowed them to place so high in the tournament. Compare GENEROUS_TIT_FOR_TAT_0.3 to TIT_FOR_TAT, for example, and consider their interactions with JOSS_0.1. Both bot pairs (GENEROUS_TIT_FOR_TAT_0.3, JOSS_0.1) and (TIT_FOR_TAT, JOSS_0.1) begin with mutual cooperation and continue until JOSS_0.1 throws its first probabilistic defection. TIT_FOR_TAT and JOSS_0.1 then alternate exploiting one another until JOSS_0.1's second probabilistic defection sets off mutual defection for the remainder of the interaction. TIT_FOR_TAT has no way of rekindling their originally mutually beneficial relationship. GENEROUS_TIT_FOR_TAT_0.3 on the other hand occasionally offers an olive branch by probabilistically cooperating after a JOSS_0.1 defection, and this can get the pair back on course for mutual cooperation. It comes at the cost of

[12] bots that do not even take their partner's moves into account, like ALL_C and RANDOM_0.8

[13] The order of cooperation rate and the order of objective success in the tournament are not perfectly equivalent within these top nine bots, however.

| Bot | Total Score | Avg Turn Score | Cooperation Rate | Good-Partner | Eigenjesus | Eigenmoses |
|---|---|---|---|---|---|---|
| EATHERLY | 53480 | 2.676 | 0.893 | 1.0 | 1.343 | 1.788 |
| MAJORITY_SOFT | 53306 | 2.667 | 0.869 | 0.947 | 1.325 | 1.793 |
| GENEROUS_TIT_FOR_TAT_0.3 | 52901 | 2.647 | 0.883 | 1.0 | 1.318 | 1.705 |
| GENEROUS_TIT_FOR_TAT_0.1 | 52805 | 2.642 | 0.829 | 1.0 | 1.285 | 1.769 |
| CHAMPION | 52740 | 2.639 | 0.884 | 1.0 | 1.338 | 1.798 |
| TIT_FOR_TAT | 51226 | 2.563 | 0.756 | 1.0 | 1.222 | 1.747 |
| MAJORITY_HARD | 51219 | 2.563 | 0.770 | 0.842 | 1.229 | 1.728 |
| TIT_FOR_TWO_TATS | 51015 | 2.553 | 0.861 | 1.0 | 1.325 | 1.831 |
| PAVLOV | 50843 | 2.544 | 0.772 | 0.842 | 1.182 | 1.470 |
| FRIEDMAN | 49926 | 2.498 | 0.608 | 0.789 | 1.048 | 1.521 |
| ALL_C | 49611 | 2.483 | 1.0 | 1.0 | 1.377 | 1.481 |
| RANDOM_0.8 | 48793 | 2.442 | 0.801 | 0.578 | 1.103 | 0.888 |
| TWO_TITS_FOR_TAT | 48664 | 2.435 | 0.651 | 0.789 | 1.105 | 1.593 |
| TESTER | 47992 | 2.402 | 0.551 | 0.684 | 0.887 | 0.768 |
| RANDOM_0.5 | 45845 | 2.294 | 0.501 | 0.473 | 0.688 | -0.009 |
| JOSS_0.1 | 42769 | 2.140 | 0.444 | 0.263 | 0.745 | 0.443 |
| SUSPICIOUS_TIT_FOR_TAT | 41450 | 2.074 | 0.467 | 0.368 | 0.780 | 0.506 |
| JOSS_0.3 | 39580 | 1.980 | 0.251 | 0.105 | 0.416 | -0.448 |
| RANDOM_0.2 | 38414 | 1.922 | 0.198 | 0.421 | 0.273 | -0.897 |
| ALL_D | 33844 | 1.693 | 0.0 | 0.052 | 0.0 | -1.481 |

Table 1: Tournament And Morality Metrics Results

GENEROUS_TIT_FOR_TAT_0.3 occasionally getting exploited more than TIT_FOR_TAT, but overall this paid off and GENEROUS_TIT_FOR_TAT_0.3 scored higher in the tournament, thanks to the exact same mechanism that caused GENEROUS_TIT_FOR_TAT_0.3 to have a higher cooperation rate.[14]

In this particular environment, EATHERLY and CHAMPION follow ALL_C in the cooperation rate ranking by adapting their retaliation rate according to their partner's cooperation rate. The cooperation rate of a bot employing this style of strategy directly depends on the cooperation rate of the other bots in the tournament, so the fact that EATHERLY and CHAMPION cooperated so often is indicative of the willingness to cooperate present in the environment. TIT_FOR_TWO_TATS, which only defects after two consecutive defections by its partner, also displayed a relatively high cooperation rate, because the bot environment had a low concentration of bots that both instigated exploitation and attempted to continue exploitation after receiving the temptation to defect.[15]

SUSPICIOUS_TIT_FOR_TAT plays the exact same strategy as TIT_FOR_TAT with the lone alteration being that it defaults to defection on the initial turn. SUSPICIOUS_TIT_FOR_TAT placed 17$^{th}$ out of 20 bots with a cooperation rate of 0.467, compared to TIT_FOR_TAT which placed 6th out of 20 bots with a cooperation rate of 0.756. In an environment of retaliatory players, first impressions can be of great significance.

## 4.2 Good-Partner Rating

Table 3 shows the bots ordered by good-partner rating.

Five of the top 6 bots in the tournament had a perfect 1.0 good-partner rating.[16] This makes it clear

| Bot | Good-Partner Rating |
|---|---|
| ALL_C | 1.0 |
| TIT_FOR_TAT | 1.0 |
| TIT_FOR_TWO_TATS | 1.0 |
| GENEROUS_TIT_FOR_TAT_0.1 | 1.0 |
| GENEROUS_TIT_FOR_TAT_0.3 | 1.0 |
| EATHERLY | 1.0 |
| CHAMPION | 1.0 |
| MAJORITY_SOFT | 0.947 |
| PAVLOV | 0.842 |
| MAJORITY_HARD | 0.842 |
| TWO_TITS_FOR_TAT | 0.789 |
| FRIEDMAN | 0.789 |
| TESTER | 0.684 |
| RANDOM_0.8 | 0.578 |
| RANDOM_0.5 | 0.473 |
| RANDOM_0.2 | 0.421 |
| SUSPICIOUS_TIT_FOR_TAT | 0.368 |
| JOSS_0.1 | 0.263 |
| JOSS_0.3 | 0.105 |
| ALL_D | 0.0526 |

Table 3: Bot List Sorted By Good-Partner Rating

[14] Of course, in a more hostile bot environment GENEROUS_TIT_FOR_TAT_0.3's generosity could end up hurting it more than helping it.

[15] PAVLOV would attempt to continue exploiting its partner once it successfully does so once, but PAVLOV is also nice (is never the first to defect) so when its partner is nice PAVLOV never even tries to take advantage of the relationship in the first place.

[16] MAJORITY_SOFT placed 2nd with a good-partner rating of 0.947, though in other runs of the same environment, MAJORITY_SOFT achieved good-partner ratings of 1.0. When paired with a RANDOM bot, a MAJORITY bot

that it is not always necessary for a bot to score higher than each partner it faces in order to succeed. In fact, it seems that any effort to do so ends up leading to worse overall performance. Many of the top placing bots in this tournament have the property that it is theoretically impossible for them to score higher than their partner in any single one of their interactions. TIT_FOR_TAT only defects once for each time its partner defects, and only after they do so, so one can see that TIT_FOR_TAT cannot possibly score more than its partner. Axelrod notes this phenomenon, saying that envy is not a good trait for an IPD strategy.[1]

It seems like a reasonable strategy to copy TIT_FOR_TAT, but try to include a mechanism for occasional exploitation where TIT_FOR_TAT is lacking (for example, against dumb bots like RANDOM or ALL_C). The JOSS strategies do just that, but they crash and burn in this tournament, ending up in the bottom 5 out of 20 bots. The JOSS strategy is essentially designed to try to come out on top of each interaction it has while still being somewhat cooperative, and this is reflected in the good-partner ratings of JOSS_0.1 (0.263) and JOSS_0.1 (0.105). The reason for JOSS strategies having low good-partner ratings seems to be the same reason they score poorly in this tournament. This is a good demonstration of the fact that it is easy to treat relationships in such a way that one always gets more out of it than the other person, but these pairwise victories are not enough to justify missing out on all the mutual benefit lost when retaliation ensues due to unnecessary instigations.

## 4.3 Eigenjesus Rating

Table 4 shows the bots ordered by eigenjesus rating. Because eigenjesus always favors cooperation, the hope is that it yields a different ranking that straight cooperation rate, so that we can see other factors come into play with, like with which types of bots the cooperations are performed. The rankings generated by the two metrics were indeed different, but as one might expect there was plenty of correlation. ALL_C is again the highest rated strategy, and EATHERLY and CHAMPION are again 2nd and 3rd, with eigenjesus ratings of 0.1.343 and 1.338,

| **Bot** | **Eigenjesus Rating** |
|---|---|
| ALL_C | 1.377 |
| EATHERLY | 1.343 |
| CHAMPION | 1.338 |
| MAJORITY_SOFT | 1.325 |
| TIT_FOR_TWO_TATS | 1.325 |
| GENEROUS_TIT_FOR_TAT_0.3 | 1.318 |
| GENEROUS_TIT_FOR_TAT_0.1 | 1.285 |
| MAJORITY_HARD | 1.229 |
| TIT_FOR_TAT | 1.222 |
| PAVLOV | 1.182 |
| TWO_TITS_FOR_TAT | 1.105 |
| RANDOM_0.8 | 1.103 |
| FRIEDMAN | 1.048 |
| TESTER | 0.887 |
| SUSPICIOUS_TIT_FOR_TAT | 0.780 |
| JOSS_0.1 | 0.745 |
| RANDOM_0.5 | 0.688 |
| JOSS_0.3 | 0.416 |
| RANDOM_0.2 | 0.273 |
| ALL_D | 0.0 |

Table 4: Bot List Sorted By Eigenjesus Rating

sometimes cooperates more than its partner, and sometimes less.

respectively. This is intuitive, because these bots defect only after their partner defects, and even then only with probability equal to their partner's defection rate, so the same bots that will cause EATHERLY and CHAMPION to not cooperate are the ones that have low eigenjesus ratings themselves and thus were not that valuable to cooperate with anyway, at least in the eyes of the eiegenjesus morality metric.

The RANDOM bots rank lower under the eigenjesus metric than under the metric of pure cooperation rate. They achieve the cooperation rates they were programmed to achieve, but they do not distinguish between kind and selfish partners, so this fixed rate of cooperation is distributed evenly across all partners, unlike most other bots in this environment who take into account the behavior of their partner. This lowers RANDOM's eigenjesus ranking because it wastes cooperations on worthless bots like ALL_D who have little to no eigenjesus rating of their own and thus do not contribute to other bots' eigenjesus ratings, while more sophisticated cooperative bots are achieving similar overall cooperation rates to RANDOM_0.8 but doing so by cooperating with the other cooperative bots and not with the uncooperative bots. In a sense, they are cooperating more efficiently with regard to eigenjesus rating than RANDOM_0.8 is able to do, and thus end up with a higher eigenjesus rating (and objective output) with a comparable amount of cooperation.

## 4.4 Eigenmoses Rating

Table 5 shows the bots ordered by eigenmoses rating.

Eigenmoses rating yielded a vastly different ranking of bots from the other metrics discussed here. First of all, TIT_FOR_TWO_TATS jumped from the 6th highest cooperation rate and the 5th highest eigenjesus rating, to the top spot with regard to eigenmoses rating, with an eigenmoses rating of 1.831. TIT_FOR_TWO_TATS correctly retaliates to the very nasty bots who have earned themselves negative eigenmoses ratings by defecting a lot, but it is not so quick to anger that it hurts its eigenmoses rating by not cooperating with bots with positive eigenmoses ratings, even though they might exploit TIT_FOR_TWO_TATS occasionally.

TIT_FOR_TAT and TWO_TITS_FOR_TAT both move up multiple spots from eigenjesus ranking to

| **Bot** | **Eigenmoses Rating** |
|---|---|
| TIT_FOR_TWO_TATS | 1.831 |
| CHAMPION | 1.798 |
| MAJORITY_SOFT | 1.793 |
| EATHERLY | 1.788 |
| GENEROUS_TIT_FOR_TAT_0.1 | 1.769 |
| TIT_FOR_TAT | 1.747 |
| MAJORITY_HARD | 1.728 |
| GENEROUS_TIT_FOR_TAT_0.3 | 1.705 |
| TWO_TITS_FOR_TAT | 1.593 |
| FRIEDMAN | 1.521 |
| ALL_C | 1.481 |
| PAVLOV | 1.470 |
| RANDOM_0.8 | 0.888 |
| TESTER | 0.768 |
| SUSPICIOUS_TIT_FOR_TAT | 0.506 |
| JOSS_0.1 | 0.443 |
| RANDOM_0.5 | -0.009 |
| JOSS_0.3 | -0.448 |
| RANDOM_0.2 | -0.897 |
| ALL_D | -1.481 |

Table 5: Bot List Sorted By Eigenmoses Rating

eigenmoses ranking, rated at 1.747 and 1.593, respectively. They are judged more favorably because of their intolerance of uncooperative bots. FRIEDMAN, with a rating of 1.521, also moves up, being rewarded for its harsh treatment of ALL_D, RANDOM_0.2, and JOSS_0.3. The opposite occurs with ALL_C, which blindly cooperates with everyone, even the likes of ALL_D.[17] ALL_C is the optimal strategy for maximizing the previously discussed morality metrics, but under eigenmoses rating ALL_C finishes at $11^{th}$, in the bottom half of the pack, with an eigenmoses rating of 1.481.

RANDOM_0.5 gets rated near 0.0, and thus contributes very little, good or bad, to any other bot's eigenmoses rating. This highlights a nice feature of eigenmoses rating, which is that when a bot is mindless and neutral, eigenmoses rating does not care how other bots treat it. And simply from a mathematical perspective, mindless and neutral seems like a natural 0 point, which gives eigenmoses rating some meaning without relying on the relative nature of the rankings. Bots rated positively by eigenmoses can be considered more moral than simply making decisions uniformly randomly, and bots rated negatively are actually more immoral than simply making decisions uniformly randomly.

# 5 Conclusion

This paper discussed a software system built to run IPD tournaments and to judge the participating bots according to various morality metric, and then analyzed the results of such runs. Instead of advocating for a certain morality metric or moral view, this paper focuses on proposing the general method and invites others to follow suit and improve upon the method shown. IPD provides a very simple model that is generalizable and extendable to real-world situations, so this method of defining morality metrics on IPD strategies could be relevant to the many questions people ask themselves about the right way to act.

As mentioned above, there are many future research directions that build on and improve this project. Here are some examples. 1) Designing and implementing more bots in order to provide a more robust strategy suite. There are countless strategies and it would raise the quality of this software system to include more programmed representations of these strategies. 2) Not only would more individual bots be of value to this project, but also constructing more bot environments would improve the robustness of the analysis. These morality metrics rely heavily on the concentrations of various types of bots, so very different results could appear just from varying the environment of participating bots. 3) Something mentioned briefly in this paper but which was not explored in detail was the idea of using the eigenvalue of the principal eigenvector to represent total goodness in the world. Formally stating this idea and putting it to the test with different bot environments would be an interesting research opportunity. 4) There are many well-documented variations on IPD. For example, evolutionary extensions of IPD are discussed in [1] and [3]. The morality metric method could be adapted to this variation on classic IPD. Another hugely important IPD variation is the addition of error and noise. This is a much better model of many real-world situations, and it changes the behavior, and thus morality, of many bots tremendously, even for small probability of error and noise occuring.

Modeling the real-world with simple games makes life strategies much easier to define and analyze. It also makes moral judgement of strategies clearer and more straightforward to reason about. This paper proposes the use of morality metrics on strategies in IPD tournaments, and invites others to build on the work done here in order to best utilize this way of thinking about philosophical questions regarding human behavior and morality.

# 6 Acknowledgments

I would like to thank my project supervisor, Scott Aaronson, for his fantastic guidance, as well as the idea, formulation, and names for eigenjesus and eigenmoses. I also give thanks to the Department of Electrical Engineering and Computer Science at MIT for providing this opportunity.

[17] ALL_D's eigenmoses rating is exactly the opposite (negative) of ALL_C's eigenmoses rating.

# Appendices

## A Software System

The code for the following system can be found and downloaded at `https://github.com/tscizzle/IPD_Morality`.

### A.1 Bot Players

The class *BotPlayer* represents the participating players in an IPD tournament. A strategy is implemented by inheriting from this class and overriding the $getNextMove()$ method, as well as adding any necessary initialization parameters. The $getNextMove()$ method takes as arguments the history of moves in the current bot interaction, the four payoff parameters, and the discount parameter $w$,[18] which is used as the probability at each turn that the interaction will end immediately. Thus, a bot can have an idea of the relative outcomes of each action combination (from the payoffs), an estimate of the length of the interaction (from $w$), and a record of its partner's behavior, and may use any subset of these to inform its strategy.[19] Strategies may include randomness.

### A.2 Arena

The class *Arena* hosts the tournaments. Its $runTournament()$ method takes in the list of participating bots, the number of meetings bots have with all other bots, the four payoff parameters, and the discount parameter $w$. First, it uses $w$ to randomly generate the lengths of the interactions the bots will have. For example, if $w = 0.9$ and the number of meetings is 3, the interaction lengths might be [7, 12, 9], meaning each bot will have an interaction of length 7 with each other bot, an interaction of length 12 with each other bot, and an interaction of length 9 with each other bot. After generating the interaction lengths, each bot is partnered with each other bot (including its own clone) for interactions of the specified lengths, and their moves and scores are saved. The results are returned as a $TournamentResults$ object.

### A.3 Tournament Results and Morality Calculations

The moves of every interaction as well as each bot's overall score are wrapped in a $TournamentResults$ object, which gets passed to a $MoralityCalculator$ object for analysis. $MoralityCalculator$ assigns scores to each bot for various morality metrics (which will be discussed in detail below).

# References

[1] Axelrod, Robert. *The Evolution Of Cooperation.* Basic Books, 1984.

[2] "Strategies for IPD." http://prisoners-dilemma.com/. Accessed April 2014.

[3] Nowak, Martin and Karl Sigmund. *A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game.* Nature, Vol 364, 1 July 1993.

[4] Rogers, Ian. "The Google Pagerank Algorithm and How It Works." http://www.sirgroane.net/google-page-rank/. 16 May 2002.

[18] $w$ can be thought of as the decay factor of the importance of each turn in an interaction.[1]

[19] The initialization parameters are assumed to be static (for example, a probability of cooperation after a certain event, or a designation of with which move to begin) so that a bot does not have a notion of internal state, and uses only the above-listed information to make its decisions.