Suspicion: Recognising and evaluating the effectiveness of extortion in the Iterated Prisoner's Dilemma

Vincent A. Knight Nikoleta E. Glynatsi

July 20, 2018

Abstract

The Iterated Prisoner's Dilemma is a model for rational and evolutionary interactive behaviour. It has applications both in the study of human social behaviour as well as in biology. It is used to understand when and how a rational individual might accept an immediate cost to their own utility for the direct benefit of another.

Much attention has been given to a class of strategies called Zero Determinant strategies. It has been theoretically shown that these strategies can "extort" any player.

In this work, an approach to identify if observed strategies are playing in an extortionate way is described. Furthermore, experimental analysis of a large tournament with 204 strategies is considered. In this setting the most highly performing strategies do not play in an extortionate way against each other but do against lower performing strategies. This suggests that whilst the theory of Zero Determinant strategies indicates that memory is not of fundamental importance to the evolution of cooperative behaviour, this is incomplete.

1 Introduction

Agent based game theoretic models have become a stalwart of the underpinning mathematics of interactive behaviours. One of the major pieces of work in this area is the pair of original computer tournaments run by Robert Axelrod [2, 3]. These tournaments pitted submitted computer strategies against each other in plays of the Iterated Prisoner's Dilemma. A common game where agents can choose to pay a slight cost to their immediate utility in the hope of building a reputation. This has been used in economic and evolutionary game theory to understand the evolution of cooperative behaviour.

Recently, a class of strategies was described in [17] that can provably extort any given opponent. In [8, 12] some questions have already been asked about the true effectiveness of these strategies in an evolutionary setting. Here another question is asked: is it possible to recognise this extortionate behaviour? A mathematical procedure for suspicion is presented: in the same way that the continued actions of an extortionate individual might raise suspicion.

This work makes use of the Axelrod Python library [11, 13] with a large number of Prisoner Dilemma strategies available to give an extensive numerical example of the ideas presented. The approach is presented in Section 2. All of the code and data discussed in Section 3 is open sourced, archived and written according to best scientific principles [22]. The data archive can be found at [10].

2 Recognising Extortion

In [17], given a match between 2 memory one strategies, the concept of Zero Determinant (ZD) strategies is introduced. The main result of that paper shows that given two memory one players $p, q \in \mathbb{R}^4$ a linear relationship between the players' scores could be forced by one of the players.

Using the notation of [17], assuming the utilities for player p are given by $S_x = (R, S, T, P)$ and for player q by $S_y = (R, T, S, P)$ and that the stationary scores of each player is given by S_X and S_Y respectively. The main result of [17] is that if

$$\tilde{p} = \alpha S_x + \beta S_y + \gamma \quad (1)$$

or

$$\tilde{q} = \alpha S_x + \beta S_y + \gamma \quad (2)$$

where $\tilde{p} = (1 - p_1, 1 - p_2, p_3, p_4)$ and $\tilde{q} = (1 - q_1, 1 - q_2, q_3, q_4)$ then:

$$\alpha S_X + \beta S_Y + \gamma = 0 \quad (3)$$

In [17] a particular type of ZD strategy is defined: extortionate strategies. If:

$$\gamma = -P(\alpha + \beta) \tag{4}$$

then the player can ensure they get a score χ times larger than the opponent. This extortion coefficient is given by:

$$\chi = \frac{-\beta}{\alpha} \tag{5}$$

Thus, if (4) holds and $\chi > 1$ a player is said to extort their opponent. Here, the reverse problem is considered: given a $p \in \mathbb{R}^4$ how does one identify α, β if they exist and is the strategy in fact acting in an extortionate way?

In this case constraints (1) and (4) correspond to:

$$\tilde{p}_1 = \alpha R + \beta R - P(\alpha + \beta) \tag{6}$$

$$\tilde{p}_2 = \alpha S + \beta T - P(\alpha + \beta) \tag{7}$$

$$\tilde{p}_3 = \alpha T + \beta S - P(\alpha + \beta) \tag{8}$$

$$\tilde{p}_4 = \alpha P + \beta P - P(\alpha + \beta) \tag{9}$$

Equation (9) ensures that $p_4 = \tilde{p}_4 = 0$. Equations (6-8) can be used to eliminate α, β, giving:

$$\tilde{p}_1 = \frac{(R - P)(\tilde{p}_2 + \tilde{p}_3)}{S + T - 2P} \tag{10}$$

with:

$$\chi = \frac{\tilde{p}_2(P - T) + \tilde{p}_3(S - P)}{\tilde{p}_2(P - S) + \tilde{p}_3(T - P)} \tag{11}$$

Given a strategy $p \in \mathbb{R}^4$ equations (9), (10-11) can be used to check if a strategy is extortionate. The conditions correspond to:

$$p_1 = \frac{(R - P)(p_2 + p_3) - R + T + S - P}{S + T - 2P} \tag{12}$$

$$p_4 = 0 \tag{13}$$

$$1 > p_2 + p_3 \tag{14}$$

The algebraic steps necessary to prove these results are available in the supporting materials.

All extortionate strategies reside on a triangular (14) plane (12) in 3 dimensions (13). Using this formulation it can be seen that a necessary (but not sufficient) condition for an extortionate strategy is that it cooperates on average less than 50% of the time when in a state of disagreement with the opponent.

As an example, consider the known extortionate strategy $p = (8/9, 1/2, 1/3, 0)$ from [19] which is referred to as `Extort-2`. In this case, for the standard values of (R, T, S, P) constraint (12) corresponds to:

$$p_1 = \frac{2(p_2 + p_3) + 1}{3} = \frac{2(1/2 + 1/3) + 1}{3} = \frac{8}{9} \tag{15}$$

It is clear that in this case all constraints hold.

This approach could in fact be used to confirm that a given strategy is acting in an extortionate manner even if it is not a memory one strategy. However, in practice, if a closed form for p is not known, then due to measurement and/or numerical error this would not work.

This problem can be written in the following linear algebraic form where $x = (\alpha, \beta)$ and $p^* = (\tilde{p}_1 - 1, \tilde{p}_2 - 1, \tilde{p}_3)$:

$$Cx = p^* \tag{16}$$

C corresponds to equations (6-8) and is given by:

$$C = \begin{bmatrix} R - P & R - P \\ S - P & T - P \\ T - P & S - P \end{bmatrix} \tag{17}$$

Note that in general, equation (16) will not necessarily have a solution. From the Rouché-Capelli theorem if there is a solution it is unique as $\text{rank}(C) = 2$ which is the dimension of the variable x. The best fitting x^* is defined by:

$$x^* = \text{argmin}_x \|Cx - p^*\|_2^2 = \sum_{i=1}^{3} ((Cx)_i - p_i^*)^2 \tag{18}$$

In the case of a system without a solution, the remaining error is referred to as SSError:

$$\text{SSError} = \|Cx^* - p^*\|_2^2 \tag{19}$$

Using equation (17) an expression for SSError and the corresponding α, β can be obtained, the algebraic steps necessary are available in the supporting materials. Using this, for a given measured set of values p^*, equations (12)-(14) correspond to:

$$\text{SSError} = \frac{(p_1^*(2P - S - T) + (R - P)(p_2^* + p_3^*))^2}{6P^2 - 4PR - 4PS - 4PT + 2R^2 + (S + T)^2} \tag{20}$$

$$p_4^* = 0 \tag{21}$$

$$p_1^* < -\frac{(p_2^* + p_3^*)(2P - S - T)}{2(P - R)} \tag{22}$$

Note that SSError, which is the square of the Frobenius norm [6], can be considered as measure of how close a strategy is to being an extortionate strategy. Suspicion of extortion then corresponds to a threshold on SSError.

By observing interactions (human or otherwise), their memory one representation can be inferred and this approach can be used to recognise extortionate behaviour. The notion of comparing theoretic and actual plays of the IPD is not novel, see for example [18]. Immediately it is noted that if the environment is noisy [23] then no strategy can be considered to be extortionate as $p_4 > 0$.

In the next section, this idea will be illustrated by observing the interactions that take place in a computer based tournament of the IPD.

3 Numerical experiments

In [19] results from a tournament with 19 strategies, was presented with specific consideration given to ZD strategies. This tournament is reproduced here using the Axelrod-Python library [11]. To obtain a good measure of the corresponding transition rates for each strategy all matches have been run for 2000 turns and every match has been repeated 60 times. All of this interaction data is available at [10]. A good match between the inferred Markov chain and the state distribution of the actual interactions has been verified. Data for this is presented in the supplementary materials.

Figure 1 shows the SSError values for all the strategies in the tournament, as reported in [19] the extortionate strategy (which has an expected SSError approximately 0) gains a large number of wins.

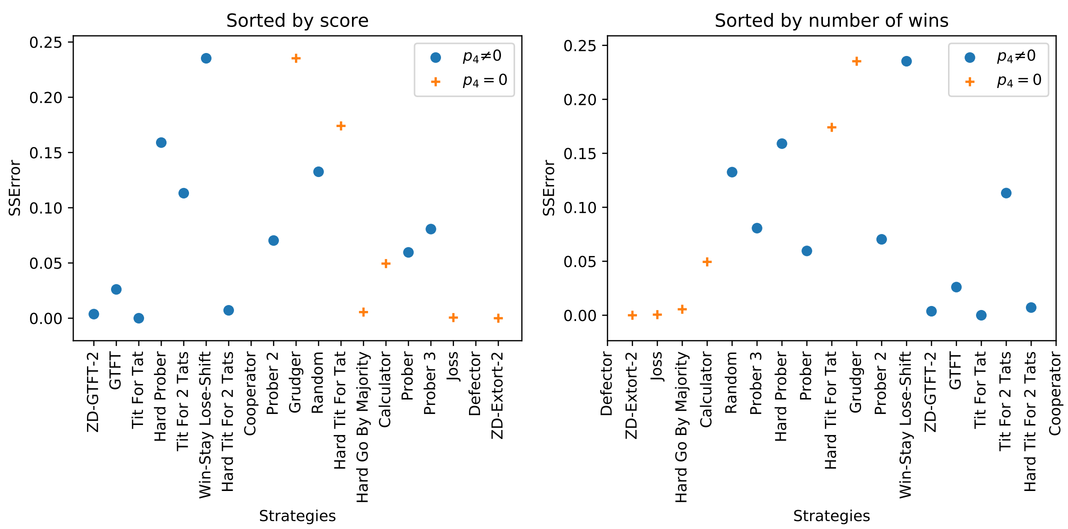

Figure 1: SSError and state probabilities for the strategies of [19], ordered both by number of wins and overall score. Note that $P(DC)$ is not shown as it corresponds to the transpose of $P(CD)$. Cooperator and Defector are omitted as they do not visit all the states.

Here, the work of [19] is extended by investigating a tournament with 204 strategies.

The results of this analysis are shown in Figure 2. The top ranking strategies by number of wins seem to be extortionate (but not against all strategies) and it can be seen that a small sub group of strategies achieve mutual defection. All the top ranking strategies according to score achieve mutual cooperation and do not extort each other, however they **do** exhibit extortionate behaviour towards a number of the lower ranking strategies.

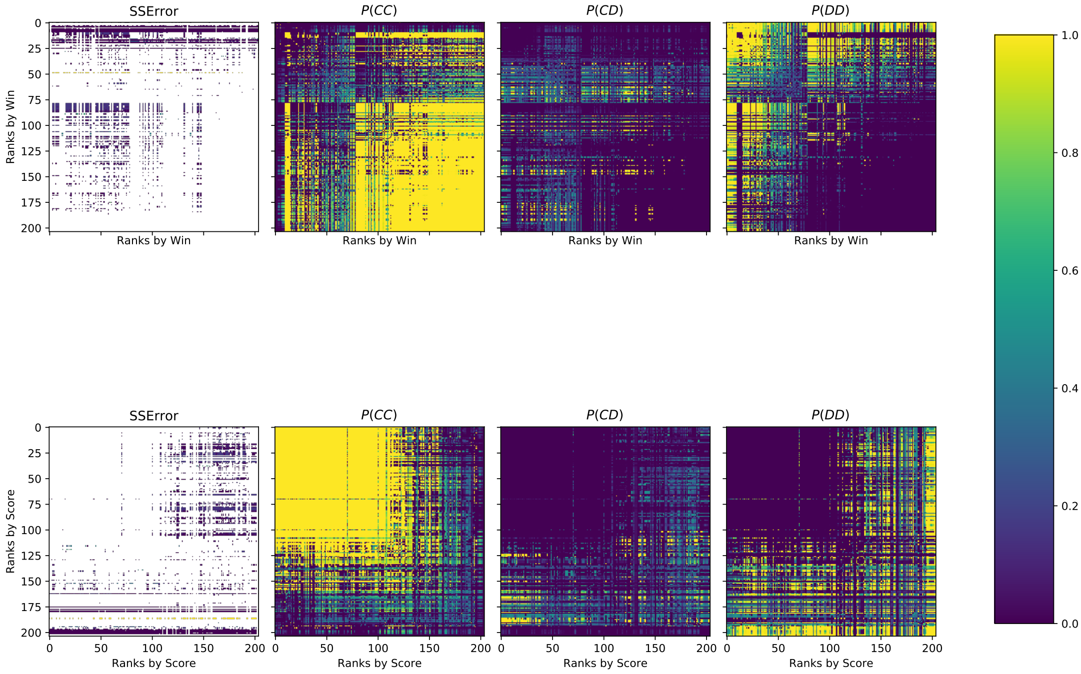

Figure 2: SSError for the strategies for the full tournament. Only strategy interactions for which $p_4 = 0$ and $\chi > 1$ are displayed.

A detailed look at selected strategies is given in Table 1. It can be seen that `Extort-2` wins many matches but does not achieve a high mean score or a high mutual cooperation rate ($P(CC)$) but it does win most of its matches.

Rank	Name	Score per turn	$P(\text{Win})$	$P(CC)$	$P(C\|CC)$	$P(C\|CD)$	$P(C\|DC)$	$P(C\|DD)$	SSError	α	β	χ
1	EvolvedLookerUp2_2_2	2.944	0.230	0.673	0.938	0.498	0.207	0.470	0.0095	0.038	-0.104	2.765
7	Evolved ANN 5	2.893	0.225	0.682	1.000	0.930	0.001	0.000	0.0011	0.001	-0.013	12.225
31	ZD-GTFT-2	2.721	0.000	0.806	1.000	0.125	1.000	0.250	0.0037	0.199	-0.176	0.889
45	ZD-GEN-2	2.689	0.016	0.801	1.000	0.562	0.500	0.125	0.0009	0.099	-0.088	0.889
69	Tit For Tat	2.638	0.000	0.723	1.000	0.000	1.000	0.000	0.0000	0.200	-0.200	1.000
88	Win-Stay Lose-Shift	2.616	0.099	0.649	1.000	0.000	0.000	1.000	0.2353	0.012	-0.188	16.000
200	ZD-Extort-2	1.821	0.851	0.179	0.889	0.500	0.334	0.000	0.0000	0.056	-0.111	1.998

Table 1: Summary of overall results for a selected list of strategies. The transition rates are computed as an average over all matches.

4 Conclusion

This work defines an approach to measure whether or not a player is playing a strategy that corresponds to an extortionate strategy as defined in [17]: a mathematical model for suspicion. Indeed, all extortionate strategies have been classified as lying on a triangular plane. This rigorous classification fails to be robust to small measurement error, thus a statistical approach is proposed. This is done through a linear algebraic approach for approximating the solution of a linear system. Using this, a large number of pairwise interactions is simulated and in fact very few strategies are found to act extortionately.

The work of [17], whilst showing that a clever approach to taking advantage of another memory one strategy exists: this is incomplete. Whilst the elegance of this result is very attractive, just as the simplicity of the victory of Tit For Tat in Axelrod's original tournaments was, it is incomplete. Extortionate strategies achieve a high number of wins but they do not achieve a high score which corresponds to the fitness landscape in an evolutionary sense. From the large number of interactions a payoff matrix S can be measured where S_{ij} denotes the score (using standard values of $(R, S, T, P) = (3, 0, 5, 1)$) of the ith strategy against the jth strategy. Using this, the replicator equation describes the evolution of the system based on a population density fitness function:

$$\frac{dx}{dt} = x(S - x^T Sx) \tag{23}$$

Equation (23) is solved numerically through an integration technique described in [16] and Figure 3 shows the evolution of the distribution of the system: the various strategies are ranked by scores. It is clear to see that only the high ranking strategies survive the evolutionary process (in fact, only 18 have a final distribution value greater than 10^{-2}). This confirms the findings of [12] in which sophisticated strategies resist evolutionary invasion of shorter memory strategies. Recalling Figure 2 this demonstrates that:

- Cooperation emerges through the evolutionary process: the high scoring strategies do not exhibit extortionate behaviour towards each other.
- Extortionate strategies do not survive the evolutionary process.

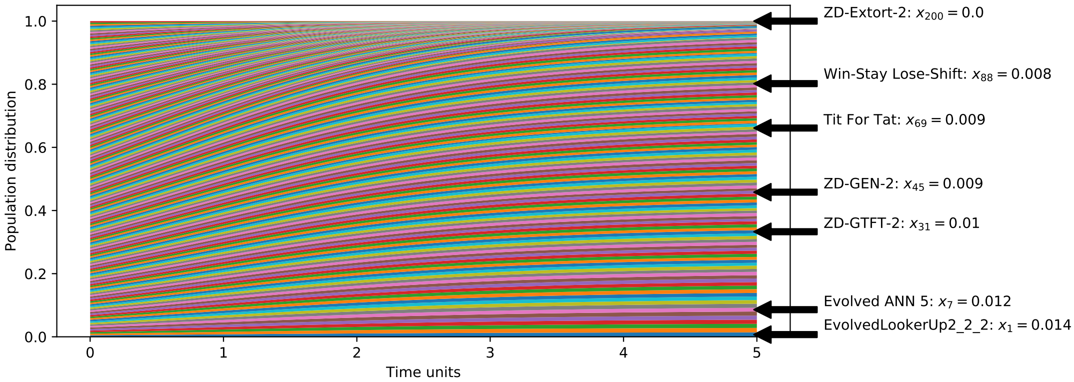

Figure 3: Numerical simulation of the replicator equation (23): strategies are ordered by score. Some selected strategies are highlighted with their long run population distribution.

This work can be used to classify plays of the IPD: data can be collected from actual interactions (in lab or in the field). Furthermore, this allows for a classification method similar to the notion of fingerprinting presented in [1]. Trained strategies can potentially be classified as extortionate or not or it could be possible to even constrain the reinforcement learning approaches that are becoming prevalent in the literature. Alternatively, this mathematical approach for recognising extortion could be used in sophisticated strategies to defend against invasion. Arguably, some of the strategies considered here exhibit this behaviour, indeed as described in [7], the top ranking strategies in the full tournament are obtained using evolutionary reinforcement learning techniques, thus, suspicion of extortionate behaviour could in fact be an evolutionary trait.

Acknowledgements

The following open source software libraries were used in this research:

- The Axelrod [11, 13] library (IPD strategies and tournaments).
- The sympy library [14] (verification of all symbolic calculations).
- The matplotlib [5] library (visualisation).
- The pandas [21], dask [4] and NumPy [15] libraries (data manipulation).
- The SciPy [9] library (numerical integration of the replicator equation).

This work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University.

References

[1] D. Ashlock and E.-Y. Kim. "Fingerprinting: Visualization and Automatic Analysis of Prisoner's Dilemma Strategies". In: *IEEE Transactions on Evolutionary Computation* 12.5 (Oct. 2008), pp. 647–659. DOI: 10.1109/tevc.2008.920675.

[2] Robert Axelrod. “Effective Choice in the Prisoner’s Dilemma”. In: *Journal of Conflict Resolution* 24.1 (Mar. 1980), pp. 3–25. DOI: 10.1177/002200278002400101.

[3] Robert Axelrod. “More Effective Choice in the Prisoner’s Dilemma”. In: *Journal of Conflict Resolution* 24.3 (Sept. 1980), pp. 379–403. DOI: 10.1177/002200278002400301.

[4] Dask Development Team. *Dask: Library for dynamic task scheduling*. 2016. URL: http://dask.pydata.org.

[5] Michael Droettboom et al. *Matplotlib/Matplotlib V2.2.2*. 2018. DOI: 10.5281/zenodo.1202077.

[6] Gene H. Golub. *Matrix Computations*. J. Hopkins Uni. Press, Jan. 7, 2013. ISBN: 1421407949. URL: https://www.ebook.de/de/product/20241149/gene_h_golub_matrix_computations.html.

[7] Marc Harper et al. “Reinforcement learning produces dominant strategies for the Iterated Prisoner’s Dilemma”. In: *PLOS ONE* 12.12 (Dec. 2017). Ed. by Yong Deng, e0188046. DOI: 10.1371/journal.pone.0188046.

[8] C. Hilbe, M. A. Nowak, and K. Sigmund. “Evolution of extortion in Iterated Prisoner’s Dilemma games”. In: *Proceedings of the National Academy of Sciences* 110.17 (Apr. 2013), pp. 6913–6918. DOI: 10.1073/pnas.1214834110.

[9] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed ¡today¿]. 2001–. URL: http://www.scipy.org/.

[10] Vincent Knight. *Raw data for: ”Suspicion: Recognising and evaluating the effectiveness of extortion in the Iterated Prisoner’s Dilemma”*. June 2018. DOI: 10.5281/zenodo.1297075. URL: https://doi.org/10.5281/zenodo.1297075.

[11] Vincent Knight et al. “An Open Framework for the Reproducible Study of the Iterated Prisoner’s Dilemma”. In: *Journal of Open Research Software* 4 (Aug. 2016). DOI: 10.5334/jors.125.

[12] Vincent Knight et al. “Evolution Reinforces Cooperation with the Emergence of Self-Recognition Mechanisms: an empirical study of the Moran process for the iterated Prisoner’s dilemma”. In: 1707.

[13] Vince Knight et al. *Axelrod-Python/Axelrod: V4.2.0*. 2018. DOI: 10.5281/zenodo.1252994.

[14] Aaron Meurer et al. “SymPy: symbolic computing in Python”. In: *PeerJ Computer Science* 3 (Jan. 2017), e103. DOI: 10.7717/peerj-cs.103.

[15] Travis E. Oliphant. *Guide to NumPy: 2nd Edition*. CreateSpace Independent Publishing Platform, 2015. ISBN: 9781517300074. URL: https://www.amazon.com/Guide-NumPy-Travis-Oliphant-PhD/dp/151730007X?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=151730007X.

[16] Linda Petzold. “Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations”. In: *SIAM Journal on Scientific and Statistical Computing* 4.1 (Mar. 1983), pp. 136–148. DOI: 10.1137/0904010.

[17] W. H. Press and F. J. Dyson. “Iterated Prisoner’s Dilemma contains strategies that dominate any evolutionary opponent”. In: *Proceedings of the National Academy of Sciences* 109.26 (May 2012), pp. 10409–10413. DOI: 10.1073/pnas.1206569109.

[18] David G. Rand and Martin A. Nowak. “Human cooperation”. In: *Trends in Cognitive Sciences* 17.8 (Aug. 2013), pp. 413–425. DOI: 10.1016/j.tics.2013.06.003.

[19] A. J. Stewart and J. B. Plotkin. “Extortion and cooperation in the Prisoner’s Dilemma”. In: *Proceedings of the National Academy of Sciences* 109.26 (June 2012), pp. 10134–10135. DOI: 10.1073/pnas.1208087109.

[20] William J. Stewart. *Probability, Markov Chains, Queues, and Simulation*. Princeton Univers. Press, July 11, 2009. 760 pp. ISBN: 0691140626. URL: https://www.ebook.de/de/product/8052317/william_j_stewart_probability_markov_chains_queues_and_simulation.html.

[21] Data Structures et al. “PROC. OF THE 9th PYTHON IN SCIENCE CONF. (SCIPY 2010)”. In: 2010.

[22] Greg Wilson et al. “Best Practices for Scientific Computing”. In: *PLoS Biology* 12.1 (Jan. 2014). Ed. by Jonathan A. Eisen, e1001745. DOI: 10.1371/journal.pbio.1001745.

[23] Jianzhong Wu and Robert Axelrod. “How to Cope with Noise in the Iterated Prisoner’s Dilemma”. In: *Journal of Conflict Resolution* 39.1 (Mar. 1995), pp. 183–189. DOI: 10.1177/0022002795039001008.

Supplementary materials

Proof of algebraic condition for extortionate strategies

The defining equations for an extortionate strategy are:

$$\tilde{p}_1 = \alpha(R-P) + \beta(R-P) \tag{1}$$
$$\tilde{p}_2 = \alpha(S-P) + \beta(T-P) \tag{2}$$
$$\tilde{p}_3 = \alpha(T-P) + \beta(S-P) \tag{3}$$
$$\tilde{p}_4 = 0 \tag{4}$$

Using equation (2, α is isolated

$$\alpha = \frac{-\beta(P-T) - \tilde{p}_2}{P-S} \tag{5}$$

Substituting this value in to equation (3), β is isolated:

$$\beta = -\frac{P\tilde{p}_1 - P\tilde{p}_2 + S\tilde{p}_2 - T\tilde{p}_1}{(S-T)(2P-S-T)} \tag{6}$$

Substituting this back in to (5) gives:

$$\alpha = \frac{-\tilde{p}_2 + (P-T)(P\tilde{p}_1 - P\tilde{p}_2 + S\tilde{p}_2 - T\tilde{p}_1)}{(S-T)(2P-S-T)(P-S)} \tag{7}$$

Substituting equations (6-7) in to equation (1) gives the required expression for p_1.
Taking the ratio of equations (6-7) gives the required expression for χ.
Finally, the condition $\chi > 1$ corresponds to:

$$\tilde{p}_2(P-T) + \tilde{p}_3(S-P) > \tilde{p}_2(P-S) + \tilde{p}_3(T-P) \tag{8}$$

which can be simplified to:

$$\tilde{p}_2 < -\tilde{p}_3 \tag{9}$$

recalling that $\tilde{p}_2 = p_2 - 1$ and $\tilde{p}_3 = p_3$ gives the required result.

Proof of optimal values of SSError

We start by obtaining an expression for SSError in terms of α, β and a measured $\|Cx - p^*\|_2^2$ where $x = (\alpha, \beta)$:

$$\|Cx - p^*\|_2^2 = (-P(\alpha+\beta) + R\alpha + R\beta - p_1^*)^2 + (-P(\alpha+\beta) + S\alpha + T\beta - p_2^*)^2 + (-P(\alpha+\beta) + S\beta + T\alpha - p_3^*)^2 \tag{1}$$

In order to find SSError we need to optimise the above in terms of α, β.
Differentiating (1), with respect to α and equating to 0 we obtain following value for α:

$$\alpha = \frac{-3P^2\beta + 2PR\beta + 2PS\beta + 2PT\beta - Pp_1^* - Pp_2^* - Pp_3^* - R^2\beta + Rp_1^* - 2ST\beta + Sp_2^* + Tp_3^*}{3P^2 - 2PR - 2PS - 2PT + R^2 + S^2 + T^2} \tag{2}$$

Differentiating (1), with respect to β and equating to 0, and substituting (2) we obtain the following value for β:

$$\beta = \frac{-3P^2p_2^* + 3P^2p_3^* + 2PRp_2^* - 2PRp_3^* - PSp_1^* + PSp_2^* - 3PSp_3^* + PTp_1^* + 3PTp_2^* - PTp_3^* - R^2p_2^* + R^2p_3^* + RSp_1^* - RTp_1^* + S^2p_3^* - STp_2^* + STp_3^* - T^2p_2^*}{(S-T)(6P^2 - 4PR - 4PS - 4PT + 2R^2 + S^2 + 2ST + T^2)} \tag{3}$$

Substituting this back in to (2) we obtain:

$$\alpha = \frac{3P^2p_2^* - 3P^2p_3^* - 2PRp_2^* + 2PRp_3^* - PSp_1^* - 3PSp_2^* + PSp_3^* + PTp_1^* - PTp_2^* + 3PTp_3^* + R^2p_2^* - R^2p_3^* + RSp_1^* - RTp_1^* + S^2p_2^* + STp_2^* - STp_3^* - T2p_3^*}{6P^2S - 6P^2T - 4PRS + 4PRT - 4PS^2 + 4PT^2 + 2R^2S - 2R^2T + S^3 + S^2T - ST^2 - T^3} \tag{4}$$

Substituting (3-4) in to (1) gives the required expression for SSError.
The condition $\chi > 1$ corresponds to $-\beta > \alpha$:

$$-\beta - \alpha = \frac{2p_1^*(P-R) + (p_2^* + p_3^*)(2P - S - T)}{6P^2 - 4PR - 4PS - 4PT + 2R^2 + S^2 + 2ST + T^2} \tag{5}$$

However, the denominator of (5) is positive as:

$$(6P^2 - 4PR - 4PS - 4PT + 2R^2 + S^2 + 2ST + T^2) = (T - 2P - S - \sqrt{2}i(P-R))(T - 2P - S + \sqrt{2}i(P-R)) \tag{6}$$

so it is a quadratic in T with no real roots and leading coefficient 1. So $\chi > 1$ corresponds to:

$$(2p_1^*(P-R) + (p_2^* + p_3^*)(2P - S - T)) > 0 \tag{7}$$

which gives to the required expression because $2(P-R) < 0$ by the definition of the Prisoner's Dilemma.

Using the pair wise interactions the transition rates p, q can be measured and the steady state probabilities inferred and compared to the actual probabilities of each state. This is done numerically by computing the singular eigenvector of the matrix A [20]:

$$A = \begin{bmatrix} p_1 q_1 & p_1(1-q_1) & (1-p_1)q_1 & (1-p_1)(1-q_1) \\ p_2 q_2 & p_2(1-q_2) & (1-p_2)q_2 & (1-p_2)(1-q_2) \\ p_3 q_3 & p_3(1-q_3) & (1-p_3)q_3 & (1-p_3)(1-q_3) \\ p_4 q_4 & p_4(1-q_4) & (1-p_4)q_4 & (1-p_4)(1-q_4) \end{bmatrix}$$

Figure 4 shows a regression line fitted to every pairwise interaction with a reported SSError value (pairwise interactions with missing states were omitted). This serves to validate the approach: a part from some edge cases the relationship is consistent.

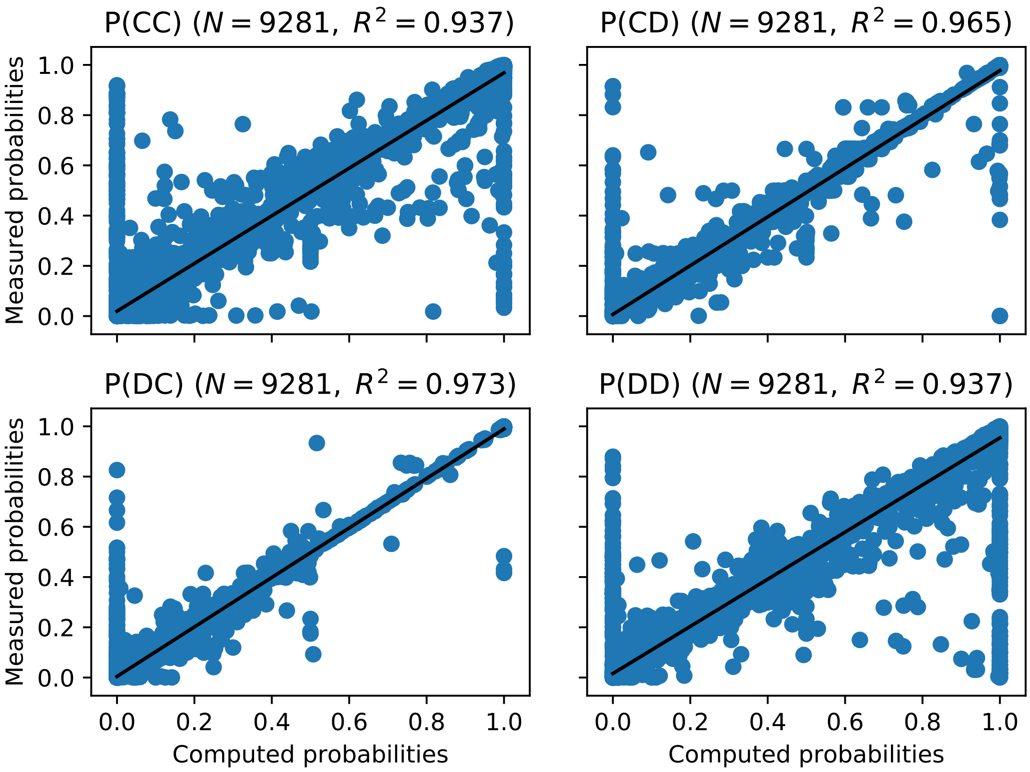

Figure 4: The relationship between the steady state probabilities inferred from the measured transitions and the actual steady state probabilities. A linear regression line is included validating the approach.