

The Structure of a Probabilistic 2-state Finite Transducer Representation for Prisoner's Dilemma

Jeffrey Tsang

Abstract—Several studies have used the fingerprint, a mathematical technique that generates a representation-independent functional signature of a game playing strategy, to conduct automated analyses of spaces of strategies. This study looks at an even larger state space, namely a grid over the probabilistic 2-state finite transducers, as a representation for playing Prisoner's Dilemma. Even using just a three-level $\{0, 0.5, 1\}$ grid amounts to 100,000 representable strategies, with an immense 40,679 unique strategies within. All strategies are fingerprinted and all pairwise distances computed, then hierarchical clustering reduces this dataset to around size 10,000 for further analysis with multidimensional scaling. Results indicate that the 20-dimensional grid has no obvious cutoff scales of structure, that we can quantify several important dimensions, and a high level of similarity with past results on smaller state spaces. We also find an interesting difference between complete playing equivalence of deterministic versus probabilistic transducers.

I. INTRODUCTION

The mathematical game is a simple to understand model for simulating interactions; however even the simplest non-trivial game, a simultaneous, symmetric two-move game, such as Prisoner's Dilemma, is difficult to understand theoretically. In studies, the game is typically iterated, allowing complex strategies of response and counter-response to your opponent. One often-used way of experimenting is with evolutionary game theory, which generates an unlimited stream of arbitrarily complicated strategies.

A series of papers [1], [2], [3] presented the concept of fingerprinting, which turns the strategies into normal mathematical functions recording the strategy's behaviour against a reference opponent, after which they become easier to handle. This has enabled previously unattainable studies in evolutionary time and population size [4], the effect of noise [5], [6] among others.

The model was updated in [7], generalizing the original one in [1], which improves upon several limitations including discontinuities and problems with indistinguishable pairs of strategies. From [8], a metric has been defined on the space of fingerprints, which allows mathematical quantification of the *distance* between particular strategies.

From this, there is a *structure* imposed on sets of strategies, especially those sets arising from spaces of representations for game-playing strategies. Several studies [9], [10] have demonstrated that the choice of representation used in an evolutionary simulation can have a drastic effect on the results, hence our interest in investigating the structure of entire representations themselves. The basic dataset we can

obtain from the fingerprint is the pairwise distance between all possible pairs of strategies in the space.

However, computing it is difficult due to the exponential number of possible strategies; with advances in technology thanks to Moore's law, we attempt here to again push the numerical size limits available. We shall look into a probabilistic 2-state finite transducer representation for Prisoner's Dilemma. There are a full 100,000 nominal strategies, and even after removing all duplicates 40,679 unique strategies remain, which is almost double the previous study [8].

Computing the distance matrix takes quadratic time; dimensionality reduction techniques can be cubic or worse. Even then, we can attempt to use as many points as currently possible to reduce any possible artifacts from clustering. We can thus consider the global structure (in the genotype space) imposed by the fingerprint distance (as phenotypic differences), as well as the mutational connectivity network.

The rest of the paper is organized as follows: the fingerprint is defined and useful properties given in Section II, the experiments are listed and described in Section III, the results and interpretation follow in Section IV, finally the discussion and conclusion are in Section V.

II. BACKGROUND

Prisoner's Dilemma is a standard two-player, two-move symmetric game studied widely in many contexts [11]. Both players on each round independently choose one of two moves: *cooperate* or *defect*. If both cooperate, each scores R ; if both defect, each scores P ; if one defects and the other cooperates, the defector scores T and the cooperator S . The conditions of Prisoner's Dilemma dictate that $T > R > P > S$ (the natural ordering of payoffs) and $2R > T + S$ (mutual cooperation is preferable to alternate backstabbing). However, our study does not directly use the payoff values.

A *probabilistic* finite state transducer (PFT) is a simple extension of the more typical deterministic finite state transducer [12]. It consists of a finite nonempty set of *states*, finite nonempty *input* and *output alphabets*, a *transition/output function* (which takes as arguments the current state and input letter, and returns a probability distribution over states and output letters), and an *initial state/output* (another distribution over states/outputs).

In our case, both input and output alphabets are the possible game moves. As a strategy, a PFT plays as follows: first *sample* the initial state and output (which move it plays) from the initializing distribution. Each subsequent round, it looks up the transition/output distribution corresponding to

Jeffrey Tsang is with the Department of Mathematics & Statistics, University of Guelph, Ontario, Canada. (email: jeffrey.tsang@ieee.org).

its current state and the opponent's last round move, and samples from that its next state and output.

As developed in [7], the fingerprint operator used in this study is based on the length-weighted probability of a move pair occurring, when the given agent plays against a parametrized k -state probabilistic finite state transducer. We will restrict our consideration to a 1-state machine, which can be parametrized as $(x, y, z) \in [0, 1]^3$, where x is the probability of cooperating on the initial move, y is the probability of cooperating in response to a cooperate, and z the probability of cooperating in response to a defect.

The operator takes as input a *specification* of a game playing agent P , which is a function ρ_P that gives the probability the agent plays as an input move history s (a string of moves) up to its length, given that its opponent plays as another input move history w (of length 1 shorter due to the simultaneity of the game) as directed. That is, $\rho_P(s, w) = \Pr(\forall i P \text{ plays } s_i \text{ in turn } i \mid \forall j \text{ opponent plays } w_j \text{ in turn } j)$. Call the parametrized opponent $O_1(\vec{v})$ with $\vec{v} = (x, y, z)$, and define $\rho_{O_1(\vec{v})}$ similarly.

Denote by \mathcal{F}_P the output of the operator on P ; the (m_1, m_2) th component of the fingerprint function is defined as $\mathcal{F}_P(\vec{v})_{m_1 m_2} =$

$$\sum_{n=1}^{\infty} \mu(n) \sum_{(s,w) \text{ has length } n-1} \rho_{O_1(\vec{v})}(wm_2, s) \rho_P(sm_1, w)$$

the first sum is the two-way probability the players play (m_1, m_2) respectively on the n th move, weighting that by a given function $\mu(n)$. For special properties, we will use the family of geometric distributions: $\mu(n; \alpha) = (1 - \alpha)\alpha^{n-1}$, $\alpha \in [0, 1)$, at $\alpha = 0.8$ in continuity with prior work [7], [8].

To compute this for agents representable by finite state transducers, create the following Markov chain: the state space is $Q \times \{C, D\}^2$, the Cartesian product of states of the agent with the last moves of P then O_1 . The transition matrix T has entries $(q_1, m_1, m_2) \rightarrow (q_2, m_3, m_4)$ equal to $\Pr(P \text{ transitions from } q_1 \text{ to } q_2 \text{ outputting } m_3 \text{ seeing } m_2) \times \Pr(O_1 \text{ outputs } m_4 \text{ seeing } m_1)$.

The fingerprint function can be calculated as

$$\mathcal{F}_P(x, y, z; \alpha)_{m_1 m_2} = (1 - \alpha) \chi_{m_1 m_2}^T (I - \alpha T(y, z))^{-1} Q_0(x)$$

where $\chi_{m_1 m_2}$ is the indicator vector whose entry is 1 if the state indexed has last move-pair (m_1, m_2) , 0 otherwise, and $Q_0(x)$ is the initial state probability vector.

Now that the strategies have been transformed into mathematical functions, we can define the distance between two fingerprints using the \mathcal{L}_1 distance (also named total variation [13]): $\|\mathcal{F}_{P_1} - \mathcal{F}_{P_2}\| =$

$$\int_{[0,1]^3} \sum_{m_1 m_2} |(\mathcal{F}_{P_1} - \mathcal{F}_{P_2})(x, y, z)_{m_1 m_2}| dx dy dz.$$

III. EXPERIMENTAL DESIGN

We shall consider a probabilistic 2-state finite transducer representation for playing iterated Prisoner's Dilemma. Since

the parameters are probabilistic and hence real numbers, the space is continuous and we must use a sampling technique. We will use a grid of 3 values $\{0, \frac{1}{2}, 1\}$ as the allowed values for all parameters.

The probabilistic transition/output (there are 5 of them: initial, state 0 or 1 in response to a cooperate or defect) is a distribution over 4 possible choices: cooperate or defect, and transition to state 0 or 1. With the aforementioned grid, there are 10 allowed choices for each distribution, and hence $10^5 = 100,000$ representable strategies.

The representation is a string of 10 characters of $\{0, 1, 2, 3\}$. 0 represents cooperate and move to state 0, 1 represents defect and move to state 0, 2 represents cooperate and move to state 1, 3 represents defect and move to state 1. Each adjacent pair of characters form both half-probability possibilities for each transition, in order of initial, state 0 in response to cooperate, to defect, state 1 in response to cooperate, to defect. We will require that the second character in each pair be no less than the first, so for example 21 is illegal (12 is equivalent).

Using standard state-minimization techniques, we can reduce this to 41,847 unique strategies under deterministic equivalence. There are 2 copies each of the true 2-state machines (as the two states can be completely mirrored, else the machine isn't actually 2-state), and 425–1192 copies of each of the 27 1-state machines. However, due to the peculiarities of probabilistic machines, there are actually only 40,679 unique strategies under full playing equivalence. This phenomenon will be discussed further.

Each of the 40,679 strategies was fingerprinted, computed using the matrix formula using the LAPACK linear algebra package into a 4-component function of y, z for $x = 0, 1$ at $\alpha = 0.8$, a value found in previous studies to have good separation properties [14].

Approximate pairwise distances are calculated with a composite third order product Gaussian cubature method (4 points at $(\pm 1/\sqrt{3}, \pm 1/\sqrt{3})$ for the region $[-1, 1]^2$, see [15]) with a grid of 512×512 evenly spaced squares (1,048,576 evaluation points). These are summed in a binary divide-and-conquer fashion to decrease roundoff error. From results in previous studies [8], the integration error is at most on the order of 10^{-10} , where distances are generally on the order of 0.01–1. For all intents and purposes, the integral is basically exact.

Hierarchical clustering with the unweighted pair group with arithmetic mean method (UPGMA, [16]) is performed on this $40,679 \times 40,679$ distance matrix. The closest pair of clusters is repeatedly merged, and the distance between two clusters is defined to be the average over all possible pairs, one from each cluster.

Dimensionality reduction methods are generally cubic or worse algorithms; hence doing it directly on a 40,679 square matrix is still infeasible. However, with ever more powerful computers, we have successfully scaled up to just under 10,000 points. That is, we pick a level of 9,952 clusters (by reversing the 9,951 largest distance mergings as done

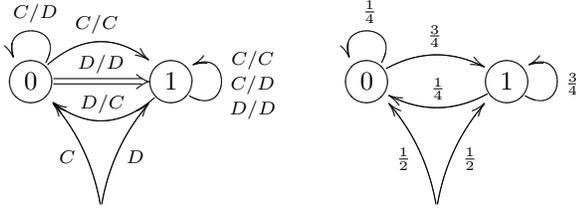


Fig. 1. The strategy represented by the string 0312332303. Left: finite state diagram. There are two half-probability possibilities for each transition; if equal they are marked with double lined arrows. Labels on the arrows are input/output; the initial transition, from below, has only output. Right: lifted transition-only graph — this is a Markov chain.

by UPGMA), and use that (weighted) distance matrix for analysis.

Metric multidimensional scaling is used to embed these clusters into the Euclidean plane. This works by minimizing the *stress* loss function

$$\sum_{i,j} w_{i,j} (\delta_{i,j} - d_{i,j})^2$$

where $w_{i,j}$ is the product of the cluster sizes of i, j , $\delta_{i,j}$ is the true distance between clusters i, j , and $d_{i,j}$ is the distance between the points on the plane representing clusters i, j .

The stress majorization SMACOF algorithm [17] is used for this purpose, with the best fit chosen from over 1,000 runs starting at initial points i.i.d. uniformly random in $[0, 1]^n$.

We can define a simple mutation operator that takes any one position in the string representation and changes it to a different value. Note that the position specifies both a transition and action; we shall require only one of them change. That is, changes have to be adjacent in the ordering 0–1–3–2–0. This induces a mutational *connectivity network* on the space that can be investigated, being an important property of the representation. We take all 100,000 strategies and find each of their 20 (not necessarily distinct) neighbours 1 mutation away, then display the propensity of a cluster to mutate into each other cluster.

IV. RESULTS

The distance matrix required almost 4 CPU-years to compute, and another 2 CPU-years was required to run the various MDS calculations.

A. Colouring

Assigning a colour to each strategy allows extra dimensions of information in a plot and perhaps we can find correlations of position to colour. We will reuse the successful schema from [8]:

Ignore the outputs for now and consider only the transition graph of the automaton, which has 2 states and 10 edges total (2 transitions from each state and the initializing transition; each has two half-probability “edges”). Run the fingerprint calculation $(1 - \alpha)(I - \alpha T)^{-1} q_0$ assigning each edge except the initializer a probability of 0.25, with the same $\alpha = 0.8$; q_0 is the distribution of initial states. This gives a visitation probability distribution on the 2 states. Multiply this by α .

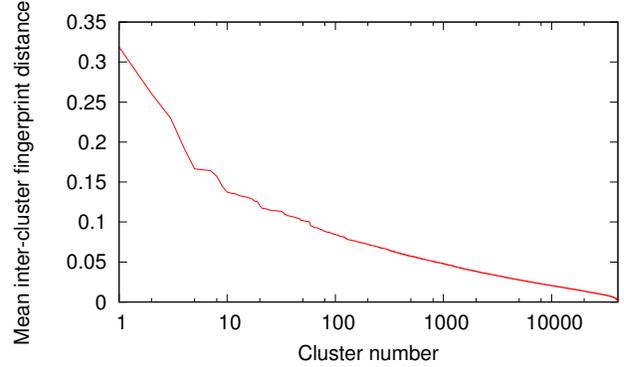


Fig. 2. The distance between clusters n and $n + 1$ when combined in the UPGMA hierarchical clustering tree. Note the logarithmic scale.

The four 1-state strategies are coloured as follows: ALLC (always cooperate) is green, ALLD (always defect) is red, TFT (tit-for-tat) is blue, and PSY (reverse tit-for-tat) is black. Consider only the actions taken while at each state, as a pair of probability distributions over cooperate/defect. Give a state a mixture colour according to the product distribution: the fraction of ALLC is the probability of cooperating in response to cooperation times the probability of cooperating in response to defect; the fraction of TFT is the probability of cooperating in response to cooperation times the probability of defecting in response to defect; and so forth for ALLD and PSY.

Now weight the states according to the above probability distribution (times α), plus the initial move coloured as ALLC/ALLD only times $1 - \alpha = 0.2$. This average is the colour of the automaton. Because the sum of all components is 1, being a probability vector, assigning black (equivalently no contribution) to PSY allows the nominally 4-dimensional surface to be fit conveniently in 3-dimensional RGB colour space.

For example, consider the strategy represented by the string 0312332303. The finite state diagram and its lifted transition-only graph is given in Figure 1.

We can calculate the visitation probability distribution of the Markov chain, which works out to be (0.3,0.7). Next we consider the action distributions at each state. From state 0, it cooperates/defects with half probability in response to cooperation, and defects in response to defect. Hence the “colour” of the state is 0.5 TFT and 0.5 ALLD. From state 1, it cooperates or defects in response to both cooperation and defection; hence its colour is 0.25 ALLC, 0.25 TFT, 0.25 PSY and 0.25 ALLD. The initial move is either cooperate or defect and hence is coloured 0.5 ALLC and 0.5 ALLD.

Weighting state 0 by $0.3 \times \alpha = 0.24$, state 1 by $0.7 \times \alpha = 0.56$ and the initial move by $1 - \alpha = 0.2$, we see that this automaton is coloured 0.24 ALLC, 0.26 TFT, 0.14 PSY and 0.36 ALLD, or (0.36,0.24,0.26) in RGB space, which is visually ■.

The colour assigned to a cluster is simply the weighted average of the colours of each constituent automaton.

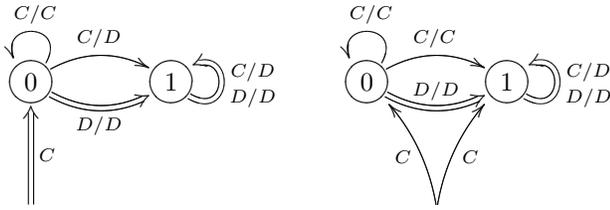


Fig. 3. Finite state diagrams for the strategies encoded by 0003333333 (left) and 0202333333 (right). There are two half-probability possibilities for each transition; if equal they are marked with double lined arrows. Labels on the arrows are input/output; the initial transition, from below, has only output.

B. Hierarchical clustering

Recall that UPGMA repeatedly joins the two clusters with the minimal inter-cluster distance; the distance between two clusters is defined to be the average over all possible pairs of points, one from the first cluster and one from the second. The distance between the clusters that are combined at each step is plotted in Figure 2. The distances do not exhibit any significant jumps other than the final few, which is to be expected. The rest of the distances are also loosely logarithmic, which suggests an even dispersion of points throughout the space.

One unexpected result, from clustering the 41,864 unique strategies under deterministic state minimization (which form the initial clusters) was finding 1,168 joinings with distances on the order of 10^{-17} . Given that the next joining was at a distance of 10^{-5} , this strongly suggests a completely different effect — that these strategies are in fact identical, even though they have different-looking representations.

We illustrate one such pair of strategies here, represented by the strings 0003333333 and 0202333333. Finite state diagrams for both are provided in Figure 3.

The first strategy is as follows: start at state 0 and cooperate first move. While at state 0, if opponent cooperates, either cooperate and stay at state 0 or defect and move to state 1 with equal probability; if opponent defects, defect and move to state 1. When in state 1 defect forever.

The second strategy reads: cooperate first move and start at either state with half probability. While at state 0, if opponent cooperates, cooperate and transition to either state with equal probability; if opponent defects, defect and move to state 1. When in state 1 defect forever.

Prima facie, these strategies clearly look different; however in the probabilistic average sense, the distributions of the these strategies' moves are identical, at each and every possible history (complete listing of opponent's past moves). We prove this directly. Both strategies cooperate first move. If the opponent ever defects, both strategies will instantly switch to unconditional defection.

Therefore we consider what these strategies do in the face of continued cooperation from the opponent. The first strategy moves to state 1 with probability 1/2 per cooperation, hence after n cooperates it is in state 0 with probability 2^{-n} . It defects with probability 1/2 in state 0 (and always from state 1), hence it defects after n cooperates with probability

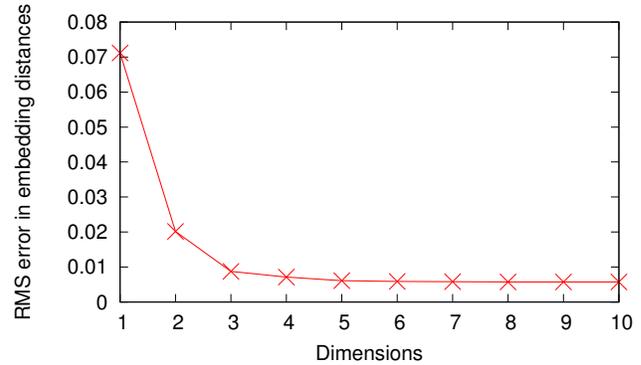


Fig. 4. The best-fit weighted root mean square error in embedding the pairwise distance matrix for the 9,952 clusters into \mathbb{R}^n , for various n . For comparison, the root mean square of the distances themselves is 0.278773.

$1 - 2^{-n-1}$. The second strategy begins in state 0 with probability 1/2, and also moves out of state 0 with probability 1/2 per cooperation: after n cooperates it is in state 0 with probability 2^{-n-1} . Since it defects if and only if in state 1, it also defects after n cooperates with probability $1 - 2^{-n-1}$!

C. Multidimensional scaling

In consideration of technical limitations, we decided to attempt to use no more than 10,000 points; we chose a cutoff at 9,952 clusters mostly arbitrarily as there are no clear levels of structure in the data. The UPGMA algorithm is re-run until 9,952 clusters remain, then the reduced distance matrix (now $9,952 \times 9,952$) is used for multidimensional scaling.

We can repeat the multidimensional scaling algorithm with different numbers of dimensions allowed for the points: clearly, the error monotonically decreases as we use more dimensions, but when the improvement starts becoming insignificant we can claim this is the essential dimension of the dataset, a procedure also known as a *scree plot* [18]. This is shown in Figure 4.

3 dimensions account for most of the data; the 4th and 5th give a small improvement, 6th a tiny one, and further contributions are negligible. A commonly quoted goodness-of-fit statistic is Kruskal's normalized stress, computed as

$$\sqrt{\frac{\sum_{i,j} (\delta_{i,j} - d_{i,j})^2}{\sum_{i,j} \delta_{i,j}^2}}$$

which is unity for the trivial solution for putting all the strategies at the same position. Stress below 0.05 is considered good [19]; our stress for embedding into \mathbb{R}^2 is 0.07235, into \mathbb{R}^3 is 0.03134, into \mathbb{R}^6 is 0.02106.

The reason the error does not converge to 0 is that the data, computed under \mathcal{L}_1 , is inherently not Euclidean and cannot be exactly embedded into Euclidean space of any dimension whatsoever. This residual error, which is relatively small, is a measure of the non-Euclideanness of our data.

A huge scatter plot in 3 dimensions is too difficult to handle, and so we choose to use the best 2-dimensional fit, knowing that there is a significant dimension in the data left

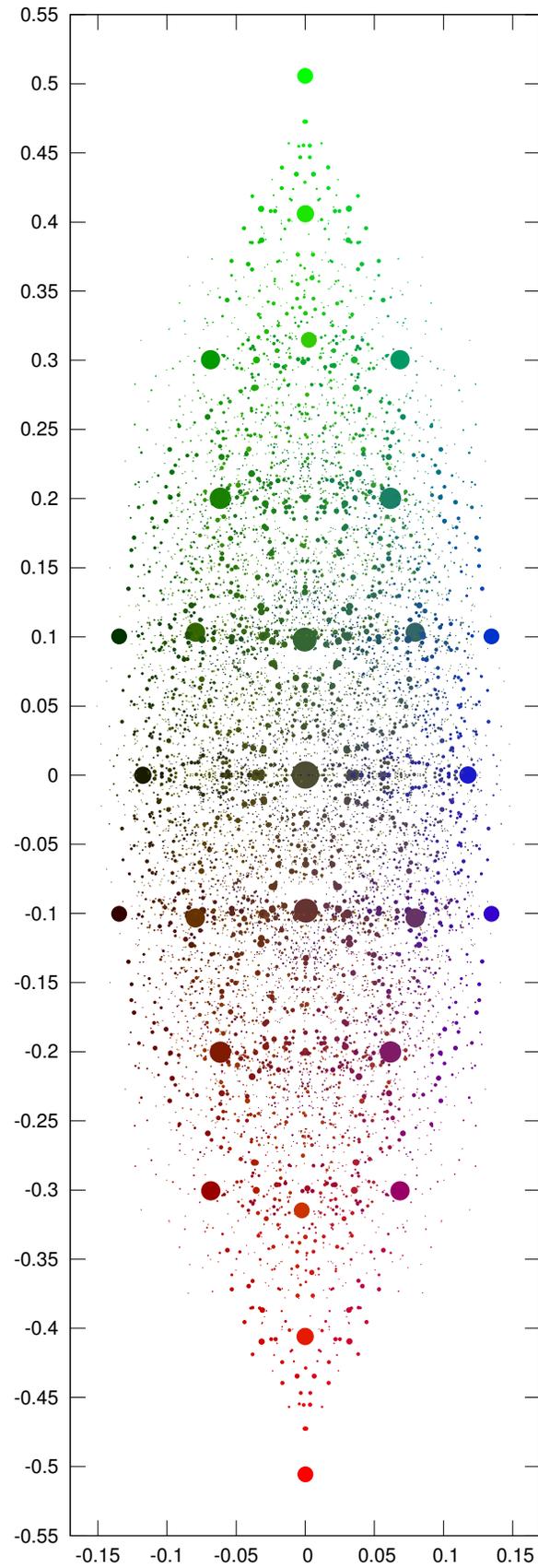


Fig. 5. Scatter plot of all 100,000 strategies, reduced to 9,952 clusters with UPGMA hierarchical clustering, projected to 2D with metric MDS. Point size (area) is directly proportional to cluster size; for colouring see section IV-A; axes are rotated to principal components, positive orientation is arbitrary.

undisplayed. These points, now explicitly in Euclidean space, are rotated to the principal components.

The clusters are plotted in Figure 5, with the colouring scheme described in Section IV-A. The first major observation from the plot is that the colouring scheme is clearly reflected in the position of the points: there is a strong correlation with the colour green with the positive y -direction, red with negative y , blue with positive x and black with negative x .

We thus confirm that the colouring scheme works. Notice also the 27 1-state strategies (which are the largest clusters) form the outlines of a cubical structure in the space, with ALLC, ALLD, TFT, D-TFT (defect first move, then Tit-for-tat), PSY, C-PSY being outer corners. The clusters corresponding to a non-deterministic parameter choice are uniformly larger due to there being more ways to generate that behaviour.

See also the clear symmetry of the configuration across both the x and y axes. Replacing an automaton with one that plays the opposite move at every step is a half-circle rotation about the origin, while reversing the responses to cooperate and defect per state is reflection across the y -axis.

D. Principal components

By comparison with [7], [14], [8] we see the high similarity in structure with results on other representations, even though these are all different representations. We thus test the hypothesis that the principal components in the embedded clusters correspond to *cooperativity* (probability of cooperating minus that of defecting) and *responsiveness* (the correlation between your move and your opponent's last move). We further make the hypothesis that the third component corresponds to *initialism* (difference in cooperativity in the first move vs. later moves).

We have a quantitative test available: we can measure the Pearson linear correlation coefficient between the colours of the clusters and their position. For our predictors, we will use linear combinations of the colours of the clusters as defined in Section IV-A. For cooperativity, we use the function Green – Red (bounded between -1 and 1), for responsiveness the predictor Blue – Black.

To predict initialism, we will separate out the normal states vs. the initial move-state, de-weight them and subtract. In the example given in Section IV-A, state 0 had weight 0.3 (we remove the weighting by α), and was coloured 0 ALLC and 0.5 ALLD (only these matter). State 1 had weight 0.7 and was coloured 0.25 ALLC and 0.5 ALLD, hence the normal-state cooperativity is 0.175 (ALLC) minus 0.5 (ALLD), or -0.325. The initial state was coloured 0.5 ALLC and 0.5 ALLD, cooperativity $0.5 - 0.5 = 0$. Thus our (unscaled) predicted value for its third coordinate is $-0.325 - 0 = -0.325$.

For the test itself, we will use the best-fit 6-dimensional MDS embedding rotated to principal components (top dimensions extremely similar to the 2-D embedding; data omitted). The Pearson correlation (bounded in [-1,1]) between Green – Red and the first principal component of the

points is 0.999164, between Blue – Black and the second component 0.951350, between the initialism predictor and the third component 0.992222. As 1 indicates a *perfect* linear relationship, these values are extraordinarily high.

E. Cluster-wise mutation

To make the data easier to display, we reduce the number of clusters to 302 the same way (removing the 301 largest distance mergings done by UPGMA). As described in Section III, each automaton has 20 neighbours in our string-based representation.

We compute for each cluster i the probability distribution of which cluster j it moves to in 1 mutation, by averaging over all automata in the cluster. This can be displayed as a heatmap in Figure 6. The ordering of the clusters is as the hierarchical clustering tree: if two clusters would be merged in the (removed) UPGMA, they are adjacent. The colour bar displays each cluster's colour and can be compared to the colours of the points in Figure 5.

One salient feature of the map is that for most clusters, the plurality (not necessarily majority) of mutations will stay within the cluster, which is not completely surprising. A key difference from comparison to [8] though, is that not only are the 1-state automata far less dominating in the space, they also no longer have an especial propensity for null mutations. This illustrates one key difference in using probabilistic automata: one main way of having a 1-state automaton is to completely not use a state given, but as more and more probability graduations are added, absolutely avoiding a state becomes a vanishing proportion of the space.

Several larger-scale squares (groups of adjacent clusters) can be discerned by their relative reachability under mutation — a lot of mutations do not move you far from the diagonal, hence to a nearby cluster. Interestingly, there is a space between the close-by clusters and the next set of clusters reachable — this is visualized by the black gap between the diagonal and the scattered outer bands.

V. DISCUSSION AND CONCLUSION

Results from hierarchical clustering show that the space of strategies is dispersed without obvious clumping. By pushing the limits of multidimensional scaling, a huge 9,952 clusters of strategies were embedded into the plane and exhibit a high degree of symmetry across both principal component axes.

The colouring scheme developed in [8], based on the fraction of time a strategy plays (for each move) as each of the 1-state strategies when playing against a RANDOM opponent, again is spectacularly successful in quantitatively *predicting* the spatial coordinates of the clusters, as evidenced by the smooth colour gradients in the scatterplot Figure 5.

We have thus quantified the top three principal coordinates of this strategy space: the first one is average probability of cooperation minus defection, the second is the correlation between your move and your opponent's last move, and the third, a novel result, is the signed difference in cooperativity between the first and later moves.

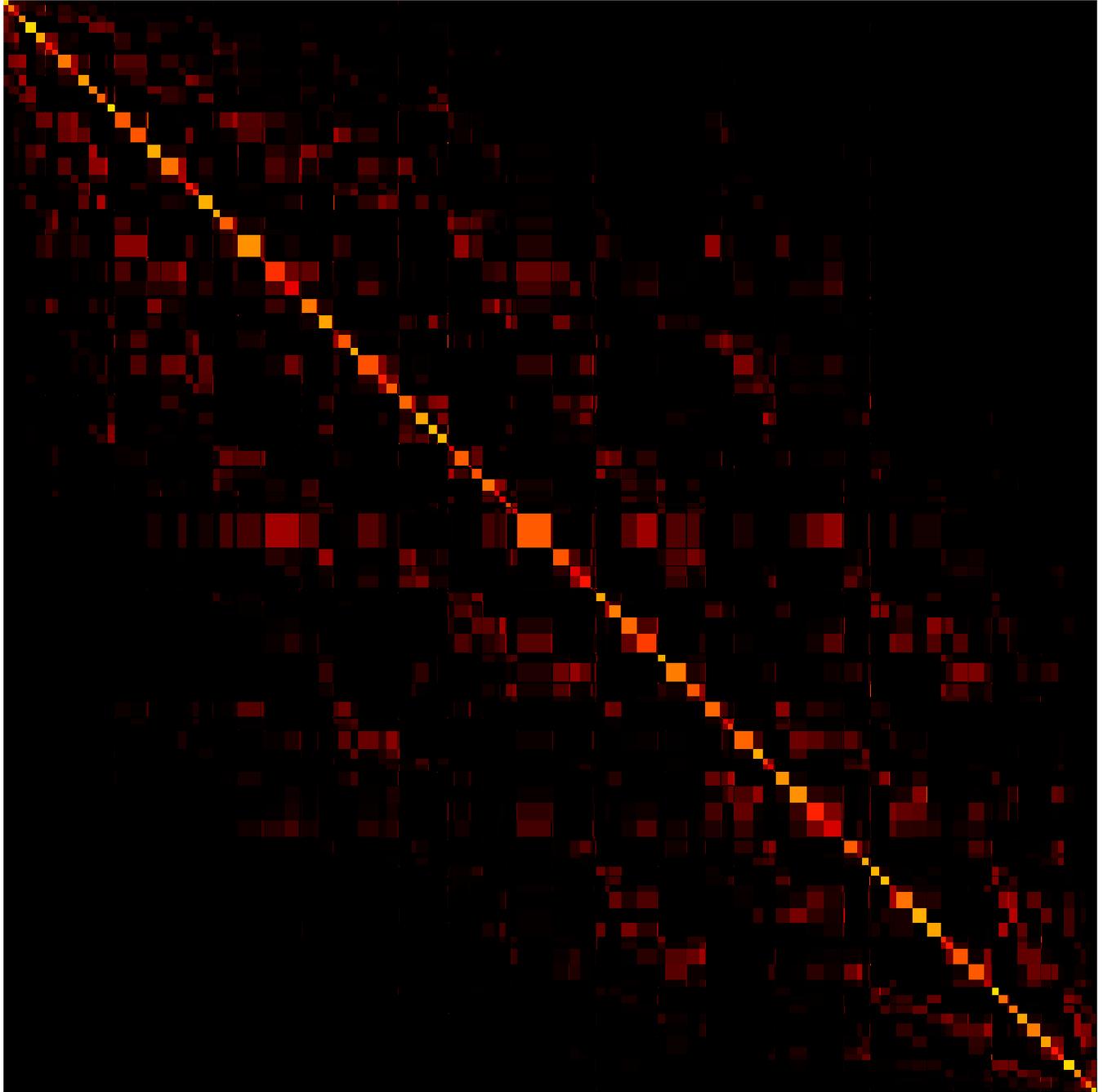


Fig. 6. Single mutation connectivity heat-map: cluster width is proportional to size and clusters are listed in the hierarchical tree order. Rectangle (i, j) has as value the (cluster average) probability of transitioning from cluster i to cluster j in one mutation: the colourmap is piecewise linear, black at 0, red at 0.25, yellow at 0.5 and white at 1.0. The colour bar on the bottom displays each cluster's colour as in section IV-A.

In conclusion, we have found several interesting pairs of automata that are indistinguishable probabilistically even though they look different in a finite state diagram, fingerprinted a massive state space and placed the points into a low dimensional space, then generated simple heuristics that explain the most important dimensions of the space. The results looks highly similar to past results on other state spaces.

Given that the state spaces in this study and [8] are very different, deterministic versus probabilistic transducers, there is no *a priori* reason that they should have such a similar structure in phenotypic (fingerprint) space; although as far as the connectivity network goes, there seems to be a larger contrast. The 1-state automata are no longer taking up a dominant segment of the representation, and most clusters have a similar propensity for null mutations.

With this level of structural similarity across disparate spaces, it is promising to ask the question: are all representations essentially similar? We can reaffirm this result be exploring even more strategy spaces, especially larger ones. At this point we can confirm that the mere presence of probabilistic transitions do not create new dimensions of difference between strategies.

The other way to encounter such extra complexity would be to add more states to the automata. Unfortunately, this directly causes the usual combinatorial explosion, therefore some sampling technique has to be used. Technically, we have already sampled to a rather sparse grid here; for even larger spaces only randomized sampling will be feasible.

On the opposite tack, we may attempt to quantify exactly how similar the current results on these spaces are. Because of their substantial overlap, particularly in that the 1-state automata are common to both, we may use them as *anchors*, and their relative movement in the MDS projections is a test against the null hypothesis that they are in the same absolute position. We may also test the new-found predictor for the third principal component on the other spaces, generating more evidence that it works on all spaces.

Another direction for future work is replicating the analysis for different choices of α , which was not done here due to computational constraints.

ACKNOWLEDGMENTS

The author would like to thank Rajesh Pereira, University of Guelph. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca) and Compute/Calcul Canada.

REFERENCES

- [1] D. Ashlock, E.-Y. Kim, and W. K. von Roeschlaub, "Fingerprints: enabling visualization and automatic analysis of strategies for two player games," in *2004 IEEE Congress on Evolutionary Computation*, Jun. 2004, pp. 381–387.
- [2] D. Ashlock and E.-Y. Kim, "Techniques for analysis of evolved Prisoner's Dilemma strategies with fingerprints," in *2005 IEEE Congress on Evolutionary Computation*, Sep. 2005, pp. 2613–2620.
- [3] —, "Fingerprinting: Visualization and automatic analysis of Prisoner's Dilemma strategies," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 647–659, Oct. 2008.
- [4] W. Ashlock and D. Ashlock, "Changes in Prisoner's Dilemma strategies over evolutionary time with different population sizes," in *2006 IEEE Congress on Evolutionary Computation*, Jul. 2006, pp. 297–304.
- [5] D. Ashlock, E.-Y. Kim, and W. Ashlock, "Fingerprint analysis of the noisy Prisoner's Dilemma using a finite-state representation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 2, pp. 154–167, Jun. 2009.
- [6] —, "Fingerprint analysis of the noisy Prisoner's Dilemma," in *2007 IEEE Congress on Evolutionary Computation*, Sep. 2007, pp. 4073–4080.
- [7] J. Tsang, "The parametrized probabilistic finite-state transducer probe game player fingerprint model," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 3, pp. 208–224, Sep. 2010.
- [8] —, "The structure of a 3-state finite transducer representation for Prisoner's Dilemma," in *2013 IEEE Conference on Computational Intelligence in Games*, Aug. 2013, pp. 307–313.
- [9] D. Ashlock, E.-Y. Kim, and N. Leahy, "Understanding representational sensitivity in the iterated Prisoner's Dilemma with fingerprints," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 36, no. 4, pp. 464–475, Jul. 2006.
- [10] D. Ashlock and E.-Y. Kim, "The impact of cellular representation on finite state agents for Prisoner's Dilemma," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Jun. 2005, pp. 59–66.
- [11] A. Rapoport and A. M. Chammah, *Prisoner's Dilemma*. Ann Arbor, MI: University of Michigan Press, Dec. 1965, vol. 165.
- [12] T. L. Booth, *Sequential Machines and Automata Theory*. New York, NY: John Wiley and Sons, 1967.
- [13] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics," *International Statistical Review*, vol. 70, no. 3, pp. 419–435, Dec. 2002.
- [14] J. Tsang, "The structure of a depth-3 lookup table representation for Prisoner's Dilemma," in *2010 IEEE Conference on Computational Intelligence in Games*, Aug. 2010, pp. 54–61.
- [15] A. H. Stroud, *Approximate Calculation of Multiple Integrals*, ser. Prentice-Hall Series in Automatic Computation. Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [16] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*, ser. A Series of Books on Biology. San Francisco, CA: W. H. Freeman, Jun. 1973.
- [17] J. de Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. R. Barra, F. Brodeau, G. Romier, and B. van Cutsem, Eds. Amsterdam, Netherlands: North-Holland Press, 1977, pp. 133–146.
- [18] D. A. Jackson, "Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches," *Ecology*, vol. 74, no. 8, pp. 2204–2214, Dec. 1993.
- [19] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964.