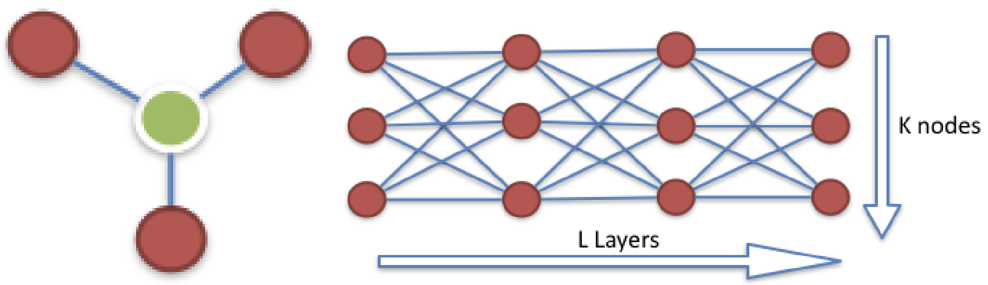
Supplementary Material for From Chaos to Order: Symmetry and Conservation Laws in Game Dynamics

Sai Ganesh Nagarajan [1] **David Balduzzi** [2] **Georgios Piliouras** [1]

A. Examples of network topologies



(a) Star graph with 3 leaf agents (indicated by red) (4-STAR) and the center agent (green).

(b) Bipartite graphs with 3 layers

Figure 1. Examples of bipartite network topologies

B. Dimensionality Reduction and Invariant Functions

B.1. Proof of Theorem 2

Proof of Theorem 2. We first use the techniques described in (Mertikopoulos et al., 2018) to define the difference in scores or payoffs, as follows. For any agent (i, j), i.e, vertex i in layer j, we may choose a benchmark strategy, wlog (say the last strategy) and then we can define the following variables:

$$z_k^{i,j} := y_k^{i,j} - y_{m-1}^{i,j} \;\; \forall k \in 0, 1, 2, \ldots, m-1 \tag{1}$$

This means that $z_{m-1}^{i,j} = 0$ and we drop this variable and we take only the first $m-1$ variables. Also , taking the partial derivative of the above equation with respect to time, we obtain:

$$\frac{dz_k^{i,j}}{dt} := u_{i,j}(k, x) - u_{i,j}(m-1, x) \;\; \forall k \in 0, 1, 2, \ldots, m-1 \tag{2}$$

In general, the above equation says that the difference in scores evolve as a function of the difference in the utilities for playing the strategy under consideration and the benchmark strategy. Important thing to note here is that, the x represents all the opponents that player (i, j) is playing against. If the there is another player in the same layer (l, j), plays a scalar multiple of the games played by (i, j), then the right hand sides for both these cases will simply be a scalar multiple of each

[1]Singapore University of Technology and Design, Singapore [2]Google DeepMind, London, UK. Correspondence to: Sai Ganesh Nagarajan <sai_nagarajan@mymail.sutd.edu.sg>, David Balduzzi <dbalduzzi@google.com>, Georgios Piliouras <georgios@sutd.edu.sg>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

other and that will allow is to find more independent invariant functions. In order to do this, we define the following:

$$w_k^{i,j} := \frac{dz_k^{i,j}}{dt} \;\; \forall k \in 0, 1, 2, \ldots, m-1 \tag{3}$$

Note that the right hand side of the equations in 2 only depends on the payoffs obtained by playing against all the connected agents which only depends on the summation $\sum_r Ax^{(r,j-1)}$.

We are going to be using the above equation to derive more invariance equations with other agents. A given layer in the bipartite graph is going to have an alternating property of row and column agents, such that if the layer 1 are row agents, then layer two are column agents and layer 3 are row agents, and so on. Thus we can define the invariant equations for each layer and we can assume that the starting layer is always has row agents and if L is odd then the last layer also has row agents and if L is even the last layer has column agents. The middle layers can be either row or column agents depending on their positions.

To obtain the reduction in the first layer, we notice that, we can compare two agents (say $(i, 1)$ and $l, 1)$ which satisfy the following equations:

$$w_k^{(i,1)} = \lambda_{(i,1)} \sum_{r=1}^{K} (Ax^{(r,2)})_k - (Ax^{(r,2)})_{m-1} \tag{First Layer Row agents}$$

$$w_k^{(l,1)} = \lambda_{(l,1)} \sum_{r=1}^{K} (Ax^{(r,2)})_k - (Ax^{(r,2)})_{m-1} \tag{4}$$

To get more independent invariant equations, we can simply consider the consecutive agents $(i, 1)$ and $(i + 1, 1)$ and apply the above equations (wlog).

Next,we assume that the last layer consists of column agents, if it consisted of row agents, the update would be similar to the above equations.

$$w_k^{(i,L)} = \sum_{r=1}^{K} \lambda_{(i,L)} (-A^T x^{(r,L-1)})_k - \lambda_{(i,L)} (-A^T x^{(r,L-1)})_{m-1} \tag{Last Layer Column agents}$$

$$w_k^{(i+1,L)} = \sum_{r=1}^{K} \lambda_{(i+1,L)} (-A^T x^{(r,L-1)})_k \tag{5}$$

$$- \lambda_{(i+1,L)} (-A^T x^{(r,L-1)})_{m-1} \tag{6}$$

Here , we can see that the invariant for column agents is simply $\lambda_{(i+1,L)} w_k^{(i,L)} - \lambda_{(i,L)} w_k^{(i+1,L)} = 0 \;\; \forall k \in 0, 1, 2, \ldots, m-2$. To get more independent invariant equations, we can simply consider the consecutive agents in the last layer and equation Last Layer Column agents will be true for all the agents in that layer.

For any middle layer $1 < j < L$ these agents act either as row agents or column agents for the layers on either side. Now again considering two consecutive agents in this layer we can write the following equations: If the middle layer has row agents then the following invariance applies

$$w_k^{(i,j)} = \lambda_{(i,j)} \sum_{r=1}^{K} (Ax^{(r,j-1)})_k - (Ax^{(r,j-1)})_{m-1} \tag{Middle Layer Row Agents}$$

$$+ \lambda_{(i,j)} \sum_{r=1}^{K} (Ax^{(r,j+1)})_k - (Ax^{(r,j+1)})_{m-1} \tag{7}$$

$$w_k^{(i+1,j)} = \lambda_{(i+1,j)} \sum_{r=1}^{K} (Ax^{(r,j-1)})_k - (Ax^{(r,j-1)})_{m-1} \tag{8}$$

$$+ \lambda_{(i+1,j)} \sum_{r=1}^{K} (Ax^{(r,j+1)})_k - (Ax^{(r,j+1)})_{m-1} \tag{9}$$

Similar to the first layer row agents, the invariant equations are simply $\lambda_{(i+1,j)}w_k^{(i,j)} - \lambda_{(i,j)}w_k^{(i+1,j)} = 0$.

If the middle layer has column agents, then the following invariance applies

$$w_k^{(i,j)} = \lambda_{(i,j)}\Bigg(\sum_{r=1}^{K}(-A^T x^{(r,j-1)})_k - (-A^T x^{(r,j-1)})_{m-1} \tag{Middle Layer Column Agents}$$

$$+ \sum_{r=1}^{K}(-A^T x^{(r,j+1)})_k - (-A^T x^{(r,j+1)})_{m-1}\Bigg) \tag{10}$$

$$w_k^{(i+1,j)} = \lambda_{(i+1,j)}\Bigg(\sum_{r=1}^{K}(-A^T x^{(r,j-1)})_k - (-A^T x^{(r,j-1)})_{m-1} \tag{11}$$

$$+ \sum_{r=1}^{K}(-A^T x^{(r,j+1)})_k - (-A^T x^{(r,j+1)})_{m-1}\Bigg) \tag{12}$$

Similar to the first layer row agents, the invariant equations are simply $\lambda_{(i+1,j)}w_k^{(i,j)} - \lambda_{(i,j)}w_k^{(i+1,j)} = 0$.

If there are totally K agents, we can obtain, for every consecutive agents $m-1$ such invariant functions (i.e, one for each strategy), or equivalently, for each strategy there are $K-1$ functions that connects the agents in the following manner:

$$\lambda_{(i+1,1)}w_k^{(i,1)} - \lambda_{(i,1)}w_k^{(i+1,1)} = 0 \;\; \forall i \in 1, 2, \ldots, K \tag{13}$$

To conclude that the aforementioned procedure allows us to obtain a valid dimensionality reduction we have to be able to recover the mixed strategies of all the agents given the initial conditions and the $L(m-1)$ variables for the L representative agents in each layer. Let us see this for one layer and show that we can indeed recover all the mixed strategies given the $(m-1)$ variables of the first agent in that layer. To see this, after integrating the invariant equations on both sides with respect to time, we obtain equations that only depend on $z^{i,j}$ and to recover the mixed strategies from $z^{(i,j)}$, we notice that $z^{(i,j)} = y^{i,j} - \mathbf{1}y_{m-1}^{i,j}$ in vector form. We need to pass this to the function Q_i to recover the mixed strategies. Applying the definition of Q on $z^{(i,j)}$ we get the following relationship

$$\begin{aligned} Q_i(z^{(i,j)}) &= \arg\max_{x_i \in \mathcal{X}_i}\{\langle z^{(i,j)}, x_i\rangle - h_i(x_i)\} \\ &= \arg\max_{x_i \in \mathcal{X}_i}\{\langle y^{i,j} - \mathbf{1}y_{m-1}^{i,j}, x_i\rangle - h_i(x_i)\} \\ &= \arg\max_{x_i \in \mathcal{X}_i}\{\langle y^{i,j} - \mathbf{1}y_{m-1}^{i,j}, x_i\rangle - h_i(x_i)\} \\ &= \arg\max_{x_i \in \mathcal{X}_i}\{\langle y^{i,j}, x_i\rangle - \langle \mathbf{1}y_{m-1}^{i,j}, x_i\rangle - h_i(x_i)\} \\ &= \arg\max_{x_i \in \mathcal{X}_i}\{\langle y^{i,j}, x_i\rangle - y_{m-1}^{i,j} - h_i(x_i)\} \\ &= \arg\max_{x_i \in \mathcal{X}_i}\{\langle y^{i,j}, x_i\rangle - h_i(x_i)\} \\ &= Q_i(y^{(i,j)}) \end{aligned} \tag{14}$$

Having h_i satisfy the above assumptions will ensure uniqueness of x_i in the strategy space for a given y_i and the last line says that if we apply Q function on $z^{(i,j)}$ we can recover the mixed strategies as if we had the information about the $y^{(i,j)}$. This final step ensures that the dimensionality reduction is valid. □

C. Reverse-Engineering the Game

C.1. Proof of Theorem 7

Proof of Theorem 7. We continue the proof of the two player-zero sum game and the multi-player case. First, in the two player game, the construction when $|S_1| = |S_2|$ is given in Section 5. Now,when they are unequal we have the following:
Case: $|S_1| \neq |S_2|$:

As we move towards more general cases (e.g. $|S_1| \neq |S_2|$) enforcing uniqueness of fully mixed equilibrium strategies is impossible. Here, we will focused on creating nontrivial zero-sum games that exhibit the desired fully mixed equilibrium strategy $(x_0, x_1 \ldots, x_r)$, $(y_0, y_1 \ldots, y_t)$ with $r \neq t$. We can assume that $r > t$, since the other case is completely symmetric. We consider the vector $(x_0, x_1, \ldots, x_{t-1}, \sum_{i=t}^{r} x_i)$. Clearly, this vector can be interpreted as a fully mixed strategy of an agent with exactly $t+1$ strategies that we have already addressed. If we pad the matrix $A(c, (x_0, x_1, \ldots, x_{t-1}, \sum_{i=t}^{r} x_i)), \mathbf{y})$ with $r-t$ extra copies of its last row, then $(x_0, x_1 \ldots, x_r)$, $(y_0, y_1 \ldots, y_t)$ is a fully mixed Nash equilibrium of this game[1] and the value of the game is equal to c. We denote this matrix again as $A(c, \mathbf{x}, \mathbf{y})$.

$$\begin{bmatrix} a(c) & \frac{c-x_1}{x_0} & \frac{c-x_2}{x_0} \cdots & \frac{c-x_t}{x_0} \\ \frac{c-y_1}{y_0} & & & \\ \frac{c-y_2}{y_0} & & I_{t\times t} & \\ \vdots & & & \\ \frac{c-y_t}{y_0} & & & \\ \frac{c-y_t}{y_0} & 0 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ \frac{c-y_t}{y_0} & 0 & \cdots & 1 \end{bmatrix}$$

Multi-agent Separable (Network) Zero-Sum: In the general case, we have n agents with arbitrary strategy sets $(S_1, S_2, \ldots, S_n)$. Once again, we wish to construct families of network zero-sum such that the target product of probability distributions corresponds to a fully mixed Nash equilibrium. We can again reduce these problem to the general two-agent case. Specifically, for any pair of agents i, k (and for any value c) we have produced a two-agent zero-sum (or equivalently a parametric family of constant-sum games) such that any target product of fully mixed strategies q_i, q_k is a Nash equilibrium of the game. It is immediate that any graphical game in which each edge corresponds to zero-sum games defined by payoff matrices $A(c, q_i, q_k)$[2] has $(q_1, \ldots, q_n)$ as (fully mixed) equilibrium. Such constructions allow us to embed long-range correlations even amongst agents that are far from each other (in terms of graph distance) via sparse graphical games with only $O(n)$ games/edges[3] instead of $\Omega(n^2)$ edges. $\square$

References

Mertikopoulos, P., Papadimitriou, C., and Piliouras, G. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2703–2717. SIAM, 2018.

[1] In fact, any strategy of the form $(x_0, x_1 \ldots, x_{t-1}, x_t, \ldots, x_r)$ is a Nash strategy for the first agent.

[2] or more generally constant-sum version of this game

[3] Each edge corresponds itself to a sparse payoff constant-sum matrix.