TOPOLOGY INFERENCE IN EVOLUTIONARY GAMES ON GRAPHS

Christopher Li[1], *Wenjun Chen*[2], *and Santiago Segarra*[3]

[1]Department of Physics, Rice University, Houston, TX
[2]Department of Physics, New York University, New York, NY
[3]Department of Electrical and Computer Engineering, Rice University, Houston, TX

ABSTRACT

We consider the task of estimating weighted adjacency matrices in evolutionary games on graphs. In evolutionary games on graphs, nodes play games with their neighbors and update their strategies based on the outcomes of these games. Existing link prediction methods often have limitations. We propose a correlation matrix designed to estimate the weighted adjacency matrix using convolutional neural networks (CNNs). Our numerical experiments illustrate that this method can predict weighted adjacency matrices with high precision. Finally, we extend the correlation matrix to correlation tensor and apply to hypergraphs.

Index Terms— Evolutionary Games, Machine Learning, Edge Weight Estimation, Correlation Matrix

1. INTRODUCTION

Ever since the seminal contributions of John Maynard Smith [1] and John Nash [2], evolutionary games have garnered significant attention. Studies have discovered that the population dynamics of certain species, such as E-coli [3] and side-blotched lizard [4], can be modeled using evolutionary game theory, owing to their phenotypes which exhibit a rock-paper-scissors dynamic. Furthermore, research [5] [6] has been conducted on evolutionary games within finite populations and those exhibiting spatial structures.

Building on the foundation laid by those pioneering studies, recent research has started to veer towards the examination of evolution games on networks [7] [8] [9]. Here, nodes within these networks adopt strategies and play games with their neighbors. Fig. 1 shows an example of an evolutionary game on a graph. Networks pervade almost every aspect of our lives, from friendship [10] to scientific collaboration [11] [12] to biological processes [13] [14]. Studies [7] [15] [16] found that spatial and network structure plays an important role in emergence of cooperation in evolutionary prisoner's dilemma games and evolutionary snowdrift games.

In practical applications, we frequently have access to an extensive volume of node evolution data, such as historical behavioral patterns of individuals [17]. On the other hand, the nature of interactions between these nodes (edge weights), especially human interactions, often remains elusive. Gaining insight into these interactions is crucial. However, several challenges complicate the estimation of edge weights. The underlying evolution dynamics, often unknown or obscured by randomness in the process, frequently hinder the identification of clear patterns. Additionally, evolutionary games on graphs tend to reach equilibria or quasi-equilibria [7] [9]. At (quasi-) equilibria, extracting edge information becomes an infeasible task. Furthermore, interactions in the real world are usually not limited to pairwise ones [18] [19] [20].

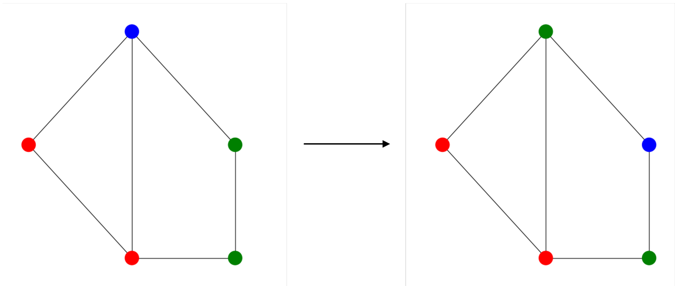

Fig. 1. A example of a rock-paper-scissors game on graphs. Red = Rock, Blue = Paper, Green = Scissors.

The majority of link prediction methods concentrate on non-weighted and dynamic networks, frequently dealing with scenarios where adjacency matrix information is only partially missing [21] [22] [23] [24] [25]. In addition, evolutionary games on higher-order graphs (hypergraphs) are not well studied.

In this work, we propose a method to predict weighted adjacency matrix from node evolutionary data on a static network. Our main contributions are as follows:

(i) We define a sample-averaged correlation matrix that can be computed from node evolution data.

(ii) We demonstrate that convolutional neural networks can effectively infer the weighted adjacency matrix from the correlation matrix.

(iii) We extend the method to higher-order graphs and show that it can accurately estimate binary hyper-edge weights.

Our approach leverages the characteristic of most evolutionary games having bounded or finite strategy sets. This method effectively addresses the previously mentioned challenges, including randomness in the evolutionary process, the existence of equilibria, and higher-order interactions.

2. PROBLEM FORMULATION

Consider a discrete-time evolution on an undirected graph $G(V,E,W)$ with nodes V, edges $E \subset V \times V$ and edge weights $W : E \to \mathbb{R}_{++}$. Here, $\mathbb{R}_{++}$ denotes all positive real numbers. We only consider graphs without self-loops. At time t, each node i has a strategy $x_i(t) \in \mathbb{S}$, where $\mathbb{S}$ is the common strategy set for all nodes. Let $|E| = N$ and the node state vector be $\vec{x}(t)$. Suppose there is an evolution rule:

$$f : \mathbb{S}^N \times {\mathbb{R}_+}^N \to \mathbb{S}, \tag{1}$$

such that $\vec{x_i}(t+1) = f\left(\vec{x}(t), A_{1i}, A_{2i}, ..., A_{Ni}\right)$, where A is the weighted adjacency matrix and $\mathbb{R}_+$ denotes all non-negative real numbers. Eq. 1 states that on each time step, each node plays games with all neighboring nodes. It updates its strategy according to the outcomes of the games and the (non-negative) edge weights associated with the games. See Section 4 for examples.

We formulate the problem as follows:

Problem Assume f is unknown and we have sufficient quantity of $\vec{x}(t=0) \to \vec{x}(t=1)$ evolution data on many known graphs. Then given sufficient quantity of $\vec{x}(t=0) \to \vec{x}(t=1)$ evolution data on an unknown graph, find an estimate of A. Here, we assume $\vec{x}(t=0)$ data is randomized.

We highlight that our choice to use evolution data from $\vec{x}(t=0)$ to $\vec{x}(t=1)$, rather than from $\vec{x}(t=0)$ to $\vec{x}(t=\infty)$, is due to the presence of (quasi-)equilibria. If only $\vec{x}(t=0) \to \vec{x}(t=\infty)$ data is available, one may pick the out-of-equilibrium part of the evolution, provided that the evolution is "random enough". However, this scenario will not be discussed in this work.

3. METHOD

The adjacency matrix represents the interaction strength between nodes. Intuitively, the larger A_{jk} is, the stronger j and k are influencing each other's strategy in the future. Motivated by this intuitive meaning, we define the correlation function that represents the extend to which node j is inflenced by node k:

$$C_{j\leftarrow k} = \left| < e^{i(s_j(t=1)-s_k(t=0))} >_{\vec{s}(t=0)} \right|, \tag{2}$$

where $< ... >_{\vec{s}(t=0)}$ means average over all possible initial conditions, s_j and s_k are the effective angles of node j and k. For non-numerical strategies, the effective angle is defined by properly distributing the strategies on $[0, 2\pi]$. For example, in rock-paper-scissors games, "rock", "paper" and "scissors" can be represented by 0, $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$; in prisoner's dilemma games, "cooperate" and "defect" can be represented by 0 and π. For numerical strategies, the effective angle is defined by

$$s(t) = 2\pi \frac{x(t) - min(\mathbb{S})}{max(\mathbb{S}) - min(\mathbb{S})} \tag{3}$$

where $x(t)$ is the numerical strategy. Here, we assume that $\mathbb{S}$ is bounded for continuous strategy sets or finite for discrete strategy sets. Intuitively, this correlation function asks the question: given a strategy $s_k(t=0)$, what would node j prefer to choose on the next time step? If there is an edge between j and k, node j should tend to choose the strategy that is the best response to $s_k(t=0)$. Otherwise, $s_j(t=1)$and $s_k(t=0)$ should be uncorrelated. As an example to illustrate the physical meaning of the correlation function, one can easily show that if $s_j(t=1) - s_k(t=0)$ is a constant, then $C_{j\leftarrow k} = 1$; and if $s_j(t=1) - s_k(t=0)$ is a random variable $Unif(0, 2\pi)$, then $C_{j\leftarrow k} = 0$.

Using Eq. 2, a correlation matrix C that includes all N^2 interactions can be easily calculated from observed node data. Here, $C_{jk} = C_{j\leftarrow k}$. We should highlight that, for a given f, C is a function of only A. In some cases, if we know f and A, C can be found theoretically. However, estimating A from only C presents a formidable challenge. Therefore, we use machine learning techniques to find an estimation function $g : C \to A$ from existing data.

We test our method with data generated via Monte Carlo simulations. Specifically, for each f, we generate a set of random graphs. Then for each f and each graph, we generate a correlation matrix by time evolving many randomized $\vec{x}(t=0)$. We use convolutional neural networks (CNNs) to learn the estimation function $g : C \to A$ and test it on another set of random graphs. The CNN models are built using TensorFlow. The input data ($N \times N$ correlation matrix) goes through two convolutional layers with kernel size $N \times N$ and "Same" padding. We use MSE loss function and the 'adam' optimizer to train the CNN.

In addition, it is apparent that while matrix A is symmetric, matrix C is not. For instance, in a star graph, the strategy of a branch node is significantly more influenced by the central node than the central node's strategy is influenced by the branch node. Although learning $g : C \to A$ with the asymmetric C is perfectly viable, we found that replacing C with $C^* = \frac{C+C^T}{2}$ enhances the learning performance. Therefore, in our numerical experiments, we use C^* to estimate A. Fig. 2 shows an example of a set of C, C^*, estimated adjacency matrix and the true adjacency matrix.

As illustrated in Fig. 3, for graphs with binary edge weights, the adjacency matrices can be inferred by applying a threshold to C^*. This process involves examining the data and selecting an appropriate threshold value. However, this approach is not effective for graphs with continuously weighted edges.

Finally, in a hypergraph, node interactions can be represented by a set of nodes $\{j, k, l\}$. Therefore, to capture the

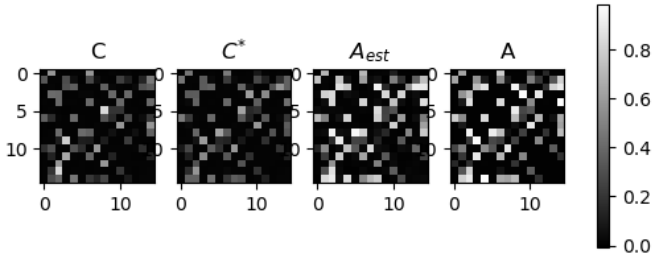

Fig. 2. An example of predicting the adjacency matrix for an Erdős–Rényi random graph in an RPS game with CNN. C is the correlation matrix, $C^* = \frac{C+C^T}{2}$, A_{est} is the estimated adjacency matrix and A is the true adjacency matrix.

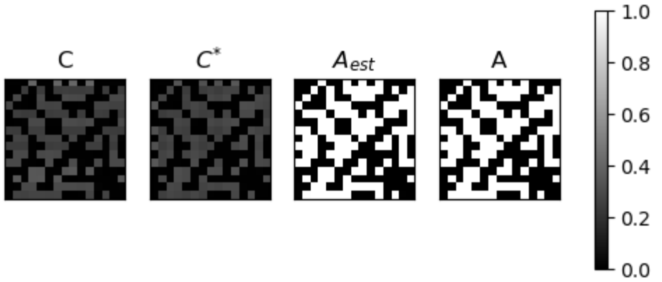

Fig. 3. An example of predicting the adjacency matrix by thresholding the correlation matrix in an Erdős–Rényi random graph with binary edge weights in an RPS game. The prediction for A_{jk} is 1 if C^*_{jk} is greater than the threshold, and is 0 otherwise.

characteristics of the interactions, we extend the definition of correlation matrix to correlation tensor. While it is difficult to find a universal correlation tensor that works for all f, we provide an example of a correlation tensor in Sec. 4, which is tailored to address a specific f.

4. NUMERICAL EXPERIMENTS

4.1. Regular Graphs

We consider three evolutionary games on star graphs, complete graphs, Erdős–Rényi (ER) random graphs and random graphs generated by preferential attachment (PA). After graphs are generated, we assign edge weights to existing edges. Each edge weight is an independent random variable $Unif(0, 1)$. The evolution games we consider are:

(I) Each node i plays rock-paper-scissors with all neighbors and earns payoff $\sum_{j=1}^{N} p_{ij}A_{ij}$, where p_{ij} is the payoff of i when playing with j. Initially, each node starts with a randomly selected strategy. After all games are played, each node i updates its strategy such that its strategy at $t + 1$ is the best response to $\vec{x}_{\bar{i}}(t)$, where $\vec{x}_{\bar{i}}$ denotes the strategies of all nodes except i. In other words, it updates strategy to maximize its payoff, assuming other nodes don't change their strategies in the next time step. We assume that players know the edge weights of all edges connecting themselves. The payoff matrix for the rock-paper-scissors game is given in table 1.

(II) Each node i plays prisoner's dilemma with all neighbors and earns payoff $\sum_{j=1}^{N} p_{ij}A_{ij}$. Initially, each node starts with a randomly selected strategy. After all games are played, each node randomly chooses a neighbor and compares their accumulated payoffs. If the neighbor's accumulated payoff is higher, the node will adopt its strategy; otherwise, it will maintain its existing strategy. The rescaled payoff matrix [6] [7] for the prisoner's dilemma game is given in table 2.

(III) Each node i plays the gold splitting game with all neighbors and earns payoff $\sum_{j=1}^{N} p_{ij}A_{ij}$. In the gold splitting game, two players are tasked with dividing 100 units of gold. Each player independently proposes a number $x \in [0, 100]$. If the sum of of their proposals exceeds 100, neither player receives anything. Otherwise, each player is awarded the amount he/she proposed. It is worth noting that the Nash equilibria of the game between two players are

$$\{(x_i, x_j)|x_i + x_j = 100\}. \tag{4}$$

After all games are played, each node i updates its strategy such that its strategy at $t + 1$ is the best response to $\vec{x}_{\bar{i}}(t)$.

	R	P	S
R	(0,0)	(-1,1)	(1,-1)
P	(1,-1)	(0,0)	(-1,1)
S	(-1,1)	(1,-1)	(0,0)

Table 1. Payoff matrix for Rock-Paper-Scissors game.

	C	D
C	(1,1)	(0,b)
D	(b,0)	(0,0)

Table 2. Payoff matrix for rescaled prisoner's dilemma game. $b > 1$. In this work, we set b=2.

For each f, we train the CNN models for each graph type (star, complete, ER or PA) and use the model to predict data for all graph types. The results are shown in Tables 3, 4 and 5. Our method has 100% accuracy on predicting binary adjacency matrices for all f's and all graph types discussed above.

4.2. 3−uniform Hypergraphs

We consider evolutionary games on 3-uniform hypergraphs, meaning each hyperedge is a set of three (and only three) nodes. We generate random hypergraphs similar to Erdos-Renyi random graphs, with each possible hyperedge has a probability 0.5 to occur. The game we consider is a 3-player

	PA	ER	Complete	Star
PA	0.00582	0.0102	0.0213	0.0131
ER	0.00897	0.00452	0.00832	0.0239
Complete	0.0797	0.0225	0.00204	0.0207
Star	0.0768	0.157	0.311	0.000535

Table 3. The Average MSE of estimations for the RPS game. Model trained with topology from the left, predicting topology from the top. Average MSE is calculated as follows: for each graph, we calculate the MSE of the estimation, then average over all testing samples. Testing sample size is 100. N=15. The probability of an edge for Erdos-Renyi graphs is 0.5. The number of new edges in each preferential attachment (Barabási–Albert model) step is 2.

	PA	ER	Complete	Star
PA	0.0122	0.0892	0.286	0.0496
ER	0.0279	0.0266	0.0767	0.155
Complete	0.616	0.174	0.0238	1.54
Star	0.0833	0.158	0.302	0.00132

Table 4. The Average MSE of estimations for the prisoner's dilemma game. Parameters are the same as table 3.

	PA	ER	Complete	Star
PA	0.00526	0.00877	0.0445	0.00745
ER	0.00568	0.00334	0.00709	0.0134
Complete	0.0173	0.00967	0.00291	0.0318
Star	0.0673	0.155	0.338	0.000693

Table 5. The Average MSE of estimations for the gold splitting game. Parameters are the same as table 3.

game with $\mathbb{S} = \{A, B, C\}$. Again, the strategies are mapped to 0, $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$. The game is won for all players when all players employ distinct strategies, whereas it is lost for all players if any strategy is duplicated among players. The payoff for winning is set to 1 for all players, while the payoff for losing is set to -2 for all players. Each node plays game within all hyperedges that contains it and accumulates payoffs. After all games are played, each node i updates its strategy such that its strategy is the best response to $\vec{x}_{\bar{i}}(t=0)$.

For this particular evolution, we propose the following problem-inspired correlation tensor:

$$C_{j\leftarrow k,l} = \left| < e^{i(s_j(t=1) - \frac{s_k(t=0)+s_l(t=0)}{2})} >_{s_k\neq s_l} \right| . \qquad (5)$$

Here, the correlation function is averaged over all initial conditions that satisfies $s_k \neq s_l$. Intuitively, the function asks the question: given three nodes, if a two nodes have difference strategies, what would the third node do? Apparently, if there is a hyperedge between the three nodes, the third node should prefer to chose the strategy that is different from the two strategies that are already present. This correlation function therefore contains information of hyperedges. One may compute all N^3 elements of the $N \times N \times N$ correlation tensor and estimate the hyperedges. Here, we used a multilayer perceptron (MLP), consists of a sequence of layers starting with a flatten layer to reshape three-dimensional input data. It goes through a dense layer with ReLU activation, followed by another Dense layer with sigmoid activation. The network uses Mean Squared Error as its loss function, and an 'adam' optimizer.

Our numerical simulation shows that the correlation tensor can predict the binary hyperedge tensor with 100% accuracy. However, we would like to highlight two key points: (A) the performance for weighted hypergraphs (not shown here) is not satisfactory, and (B) our definition of the correlation function relies on prior knowledge of the underlying evolution dynamics.

5. CONCLUSION

We have defined a correlation matrix as a tool to learn the weighted adjacency matrix from evolution data. Our numerical experiments have shown that the correlation matrix serves as an excellent tool for estimating the weighted adjacency matrix, without prior knowledge of the underlying evolution dynamics.

Potential directions for future work include investigating directed graphs and non-symmetric games, learning the weighted adjacency matrix from $\vec{x}(t=0) \to \vec{x}(t=\infty)$ evolution data and finding a general correlation tensor for evolutionary games on hypergraphs. In the presence of equilibria, one may consider analyzing only the out-of-equilibrium part of the evolution. Furthermore, clique expansion could be considered to study evolutionary games on hypergraphs.

6. ACKNOWLEDGMENT

We thank Iason Krommydas for helpful discussions.

7. REFERENCES

[1] John Maynard Smith, "Evolution and the theory of games," *American scientist*, vol. 64, no. 1, pp. 41–45, 1976.

[2] John F Nash Jr, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[3] Benjamin C Kirkup and Margaret A Riley, "Antibiotic-mediated antagonism leads to a bacterial game of rock–paper–scissors in vivo," *Nature*, vol. 428, no. 6981, pp. 412–414, 2004.

[4] Suzanne H Alonzo and Barry Sinervo, "Mate choice games, context-dependent good genes, and genetic cycles in the side-blotched lizard, uta stansburiana," *Behavioral Ecology and Sociobiology*, vol. 49, pp. 176–186, 2001.

[5] Martin A Nowak, Akira Sasaki, Christine Taylor, and Drew Fudenberg, "Emergence of cooperation and evolutionary stability in finite populations," *Nature*, vol. 428, no. 6983, pp. 646–650, 2004.

[6] Martin A Nowak and Robert M May, "Evolutionary games and spatial chaos," *nature*, vol. 359, no. 6398, pp. 826–829, 1992.

[7] Francisco C Santos and Jorge M Pacheco, "Scale-free networks provide a unifying framework for the emergence of cooperation," *Physical review letters*, vol. 95, no. 9, pp. 098104, 2005.

[8] György Szabó and Gabor Fath, "Evolutionary games on graphs," *Physics reports*, vol. 446, no. 4-6, pp. 97–216, 2007.

[9] Jiangjiang Cheng, Wenjun Mei, Wei Su, and Ge Chen, "Evolutionary games on networks: Phase transition, quasi-equilibrium, and mathematical principles," *Physica A: Statistical Mechanics and its Applications*, vol. 611, pp. 128447, 2023.

[10] Mark Newman, *Networks*, Oxford university press, 2018.

[11] Mark EJ Newman, "The structure of scientific collaboration networks," *Proceedings of the national academy of sciences*, vol. 98, no. 2, pp. 404–409, 2001.

[12] Albert-Laszlo Barabâsi, Hawoong Jeong, Zoltan Néda, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical mechanics and its applications*, vol. 311, no. 3-4, pp. 590–614, 2002.

[13] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[14] Michelle Girvan and Mark EJ Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[15] Aming Li, Lei Zhou, Qi Su, Sean P Cornelius, Yang-Yu Liu, Long Wang, and Simon A Levin, "Evolution of cooperation on temporal networks," *Nature communications*, vol. 11, no. 1, pp. 2259, 2020.

[16] Christoph Hauert and Michael Doebeli, "Spatial structure often inhibits the evolution of cooperation in the snowdrift game," *Nature*, vol. 428, no. 6983, pp. 643–646, 2004.

[17] Nathan Eagle, Alex Pentland, and David Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the national academy of sciences*, vol. 106, no. 36, pp. 15274–15278, 2009.

[18] Austin R Benson, David F Gleich, and Jure Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[19] Jacopo Grilli, György Barabás, Matthew J Michalska-Smith, and Stefano Allesina, "Higher-order interactions stabilize dynamics in competitive network models," *Nature*, vol. 548, no. 7666, pp. 210–213, 2017.

[20] Austin R Benson, Rediet Abebe, Michael T Schaub, Ali Jadbabaie, and Jon Kleinberg, "Simplicial closure and higher-order link prediction," *Proceedings of the National Academy of Sciences*, vol. 115, no. 48, pp. E11221–E11230, 2018.

[21] David Liben-Nowell and Jon Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, 2003, pp. 556–559.

[22] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 553, pp. 124289, 2020.

[23] Xu-Wen Wang, Yize Chen, and Yang-Yu Liu, "Link prediction through deep generative model," *Iscience*, vol. 23, no. 10, 2020.

[24] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy, "Local probabilistic models for link prediction," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 322–331.

[25] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos, "Edge weight prediction in weighted signed networks," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 221–230.