Which Types of Learning Make a Simple Game Complex?

Shohei Hidaka*
Takuma Torii
Akira Masumi
School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, Japan 923-1292
**shhidaka@jaist.ac.jp*

The present study focuses on a class of games with reinforcement-learning agents that adaptively choose their actions to locally maximize their rewards. By analyzing a limit model with a special type of learning, previous studies suggested that dynamics of games with learners may become chaotic. We evaluated the generality of this model by analyzing the consistency of this limit model in comparison with two other approaches, agent-based simulation and the Markov process model. Our analysis showed inconsistency between the limit model and two other models with more general reinforcement learning. This suggests that reinforcement learning does not lead to complex dynamics in games with learners.

1. Introduction

Dynamic games among agents have been investigated as minimal models of various types of social phenomena. Conventional game theory treats each agent as a rational decision maker whose actions are based on sufficient information regarding the game. The dynamics in a game with two such rational agents can be described with Nash equilibria, in which none of the agents can profit by changing their actions [1]. Real social problems are, however, often more complex than games with such rational agents [2–4]. Due to this complexity, the classic equilibria-based description may not be characteristic of realistic problems. Therefore, more recent studies have focused on games with nonrational agents that are allowed to have limited computational resources or information.

In reality, each agent has limited computational resources and information about the game. Under such uncertainty, learning plays a crucial role in finding a locally optimal action from the limited information sources. One key question regarding such games is how learning changes the dynamics [3].

Complex Systems, **24** © 2015 Complex Systems Publications, Inc.
https://doi.org/10.25088/ComplexSystems.24.1.49

As a form of minimal model with learning, a class of iterative games with *reinforcement learning* [5] has been investigated in both theoretical [6, 7] and empirical studies [2–4]. For the remainder of this work, we refer to this class of games as the *learning-and-game* model. In the learning-and-game model, each agent is typically supposed to know only a series of its own actions and the rewards for these actions. The probabilities for the agents' next actions are computed according to the weighted averages of rewards for the possible actions.

One major theoretical question regarding this model is how this local learning by each agent affects the dynamics of the game as a whole. As the learning-and-game model is essentially a stochastic process, analysis often focuses on a certain type of statistics rather than individual cases. As described in conventional game theory, the simplest class of dynamics in a learning-and-game model is convergence to a fixed point. This fixed point is the limit resulting from a certain average across agents over sufficiently many steps.

Past studies have discussed more generic classes of dynamics, which a certain type of learning-and-game model can demonstrate. Sato and colleagues have considered a limit in which learning is based on errorless feedback after infinitely many actions and rewards. They derived a set of dynamical equations of the games [7–9]. Under this limit, reinforcement learning leads from a simple game, such as rock-paper-scissors, to generic classes of dynamics including periodic, quasiperiodic, and chaotic dynamics. From these results, Sato and colleagues claimed that learning can lead a game to chaotic dynamics, to which rationality in classical game theory cannot lead. The capability of showing these complex dynamics suggests that this class of learning may be computationally universal in the sense of Wolfram [10].

However, this finding requires further investigation, as the models in [7–9] include assumptions about a special type of learning. Generally, learning should benefit one's choices under uncertainty. In contrast, Sato's continuous-time limit model assumes a special situation with no uncertainty, in which learning benefits little. Thus, it is unclear whether the generic dynamical classes can be found in more general situations. This study investigates this issue by comparing the continuous-time limit model to multiple models based on a more general learning scheme. Here, we employed agent-based simulation and Markov process formalism, each of which allows us to analyze learning-and-game models without the special limit assumption.

Agent-based simulation is one of the most common methods for analyzing learning-and-game models. In a typical agent-based simulation, a set of statistics of the agent behaviors is obtained by repeatedly applying a rule set governing each agent. One of its known shortcom-

https://doi.org/10.25088/ComplexSystems.24.1.49

ings is, however, that the sample statistics obtained in this way may not reflect the central characteristics of interest. In fact, a game with reinforcement learners heavily weighting on their past experience tends to show severe initial value dependence [11], which makes the game dynamics difficult to describe with a small number of samples. This is the situation: the specific settings leading to chaotic dynamics in [7–9] cause the convergence of the corresponding agent-based simulation to be slow. As we will discuss later, the learning-and-game model with the special settings of interest increase the difficulty of numerical computation. Given this technical problem, we need to compare multiple methods based on different assumptions and analyze the consistencies between them.

Another method we employed was a Markov process analysis. Consider an iterated game with two agents; each of them probabilistically chooses one out of a finite number of actions based on the k past steps of the game.

The probabilistic nature of this game can be sufficiently characterized by a k^{th}-order finite-state Markov process. Few past studies, however, have employed Markov process analysis on the game due to its computational cost, except for limited cases [12]. In general, a Markov process formalism requires an exponentially large number of states as k increases, and is computationally intractable.

Our motivation to employ the Markov process comes from the observation that we can relax this computational problem. Although its computational cost potentially grows as a function of k, the depth of the game tree, the Markov process analysis does not require too large a k to be a reasonable approximation to its true model. We will elaborate on this observation later.

In Section 2, we give the formal definition of the learning-and-game model, and we provide brief illustrations of the three approaches introduced. In Section 3, we numerically study the game analyzed by Sato and colleagues [9] in order to evaluate the generality of their findings. In Section 4, we study another case of the learning-and-game model for further support of our analysis in Section 3.

2. Formulation

2.1 The Learning-and-Game Model

We define the learning-and-game model in a general form here and analyze two games of this class in later sections.

Definition 1 (*n*-agent and *k*-step learning-and-game model). Consider a game with n agents, where each of the agents is denoted by integers

https://doi.org/10.25088/ComplexSystems.24.1.49

1, 2, … , n. At step $t \in \mathbb{Z}$, the i^{th} agent chooses an action $x_{t,i}$ from the set of actions $\mathcal{M} := \{0, 1, \dots, m-1\}$. Denote each of m^n possible states in n agents' actions at step t by an integer

$$X_t := 1 + \sum_{i=0}^{n-1} x_{t,n-i}\, m^{i-1}. \tag{1}$$

We write $X_t^{t+k} = (X_t, X_{t+1}, \dots, X_{t+k})$ for a sequence of k states from the step t. Define a map from a state X to a reward r for the i^{th} agent as

$$r_i : X \mapsto r \in \mathbb{R}.$$

We write the map $r_i(X_t) = r_i\left(x_{t,1}, \dots, x_{t,n}\right)$. Although this reward map may vary across different steps t in general, the present study considers only maps that remain constant across steps.

In reinforcement learning, each agent chooses an action based on a function of rewards for its past actions. This function is the weighted sum of rewards for the action $x \in \mathcal{M}$ of the i^{th} agent over k past steps

$$\phi_{i,x}^{\alpha_i}\left(X_{t-k}^{t-1}\right) = \sum_{s=1}^{k} \alpha_i^s\, \delta_{x,x_{t-s,i}}\, r_i\left(X_{t-s}\right),$$

where $\alpha_i \in [0, 1]$ is the memory-retention parameter, and $\delta_{x,y} = 1$ if $x = y$; otherwise $\delta_{x,y} = 0$. Using this weighted rewards function and the sensitivity parameter $\beta_i \geq 0$, the probability for the i^{th} agent to choose the action x at step t is

$$P\left(x \,\middle|\, X_{t-k}^{t-1}\right) = \frac{\exp\left(\beta_i\, \phi_{i,x}^{\alpha_i}\left(X_{t-k}^{t-1}\right)\right)}{\sum_{x=0}^{m-1} \exp\left(\beta_i\, \phi_{i,x}^{\alpha_i}\left(X_{t-k}^{t-1}\right)\right)}. \tag{2}$$

Assuming independent choices by the n agents, the probability of state transition is

$$P\left(X_t \,\middle|\, X_{t-k}^{t-1}\right) = \prod_{i=1}^{n} P\left(y_{t,i} \,\middle|\, X_{t-k}^{t-1}\right), \tag{3}$$

where $y_{t,i} = \left\lfloor (X_t - 1) \,\middle/\, m^{n-i+1} \right\rfloor \pmod m$, which is the inverse map of equation (1) from X_t to $y_{t,i}$.

https://doi.org/10.25088/ComplexSystems.24.1.49

2.2 Agent-Based Simulations

We define the agent-based simulation as a type of Monte Carlo simulation, using equation (3) as the probability distribution to draw a sample. We start with a series of k states $1 \le X_0, X_1, \ldots, X_{k-1} \le m^n$ and generate a value for X_{t+1} sampled from the probability distribution, equation (3), given $X^t_{\max(0,t-k+1)}$ for $t \ge k$.

The present study analyzed the case with $k \to \infty$, in which it is impossible to sample a series $X^t_{-\infty}$, and thus a truncated series $X^t_{\max(0,t-k+1)}$ must necessarily be sampled instead. This may cause biased results in this agent-based simulation, particularly as the dynamics of interest are sensitive to initial states.

2.3 Finite-State Markov Process

For $N \in \mathbb{N}$, denote the state at time $t \in \mathbb{Z}$ by $X_t \in \mathcal{N} = \{1, 2, \ldots, N\}$. We call a stochastic process the k^{th}*-order Markov process* if the probability of a state in a system at any time step is determined only by its k past states:

$$P(X_t \mid X_{t-1}, X_{t-2}, \ldots, X_{t-k}) = P(X_t \mid X_{t-1}, X_{t-2}, \ldots).$$

Definition 1 is a k^{th}-order Markov process with $N = m^n$ states.

A k^{th}-order Markov process is described by a probability vector $p \in \mathbb{R}^{N^k}$ over the joint states $1 \le i \le N^k$ and its corresponding *transition matrix* $Q \in \mathbb{R}^{N^k \times N^k}$, defined as follows.

Assign an integer $1 \le i \le N^k$ to each fixed t and joint states X^t_{t-k+1} by the indexing map

$$h_{N,k}\left(X^t_{t-k+1}\right) := 1 + \sum_{j=0}^{k-1} \left(X_{t-k+1+j} - 1\right) N^j. \tag{4}$$

Denote the inverse of the indexing map, equation (4), by $h^{-1}_{N,k}$. The probability vector $p \in \mathbb{R}^{N^k}$ over the joint states $1 \le i \le N^k$ is

$$p = \left(P\left(h^{-1}_{N,k}(1)\right), P\left(h^{-1}_{N,k}(2)\right), \ldots, P\left(h^{-1}_{N,k}\left(N^k\right)\right)\right)^T,$$

where the superscript T denotes transpose of the vector or matrix. For every $1 \le j \le N^k$, write

$$\mathcal{H}_j := \{h_{N,k}((X_1, X_2, \ldots, X_k)) : \\ j = h_{N,k}((X_0, X_1, \ldots, X_{k-1})), X_k \in \mathcal{N}\}.$$

https://doi.org/10.25088/ComplexSystems.24.1.49

For integers $1 \le j \le N^k$, the transition matrix Q corresponding to p is defined as

$$Q_{i,j} := P\left(h_{N,k}^{-1}(i) \,\middle|\, h_{N,k}^{-1}(j)\right) \tag{5}$$

for $i \in \mathcal{H}_j$ and $Q_{i,j} := 0$ for $i \notin \mathcal{H}_j$.

Suppose we start with some initial probability vector p_0. Then p_t is obtained by $p_t = Q\,p_{t-1}$ for $t > 0$. Applying this infinitely many steps, we obtain the *stationary probability distribution*

$$p_\infty = \lim_{t\to\infty} Q^t\,p_0, \tag{6}$$

if this limit exists.

The Perron–Frobenius theorem [13] gives the condition for the existence of the limit in equation (6), which we assume in this study unless otherwise specified. For a general learning-and-game model, except for those with some $\alpha_i = 1$ and $k \to \infty$, a transition matrix Q has its unique largest eigenvalue 1 and its corresponding eigenvector θ. Then the stationary distribution is given by the eigenvector of the largest eigenvalue

$$p_\infty = \theta\left(\mathbf{1}_{N^k}^T\,\theta\right)^{-1},$$

where $\mathbf{1}_{N^k} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{N^k}$ is a vector with all elements equal to 1.

2.3.1 Theoretical Properties of the Markov Process Approach

The number of states N^k of a k^{th}-order Markov process increases exponentially as the depth of the game tree k increases. This is an apparent computational difficulty. Specifically, a naive computation of the eigenvector p_∞ of the transition matrix $Q \in \mathbb{R}^{N^k \times N^k}$ in equation (5) requires matrix multiplication $Q\,p$ of the computational cost $O\left(N^{3\,k}\right)$.

However, we can compute the above multiplication more efficiently. In general, the $N^k \times N^k$ transition matrix Q is sparse, with at most N^{k+1} nonzero positive elements. Exploiting this sparsity, the matrix multiplication cost is reduced to $O\left(N^{k+1}\right)$. Further details are given in Appendix B.

An additional property is that a learning-and-game model is well approximated in general with a k^{th}-order Markov process where k is reasonably small. Consider the stationary vector $p_0 \in \mathbb{R}^{N^k}$ of a k^{th}-order Markov process with a transition matrix Q_0, and the stationary

https://doi.org/10.25088/ComplexSystems.24.1.49

vector $p \in \mathbb{R}^{N^k}$ of a transition matrix Q sufficiently close to Q_0. The sum of squared errors

$$\epsilon^2 = (p - p_0)^T (p - p_0) \approx O\left(e^{-c\,k\,\Delta}\right),$$

is an exponential function of k, where c is a constant, and $\Delta = Q - Q_0$ is the difference in the transition matrices. This suggests that a k^{th}-order Markov process exponentially approaches the true model as $k \to \infty$ for a learning-and-game model with $\alpha_i < 1$ for every i. Further mathematical elaboration is given in Appendix A. This error behavior is numerically analyzed later.

2.3.2 Special Case $\alpha_i = 1$

In the special case $\alpha_i = 1$ for each i, we can compute a stationary distribution even more efficiently. Exploiting the exchangeability of actions in a state series, the size of the state space of this special case is $(k+1)\,k \ldots (k-N+3)$ for $N \geq 2$, which is quite a bit smaller than the size of the original state space, N^k. With the following formulation, we can compute the stationary distribution for a relatively large k.

With $\alpha_i = 1$ for each i, the order of joint states in a series is inconsequential, because reward weights are equal at every step. In this case, we can identify two joint states $(X_t, X_{t-1}, \ldots, X_{t-s}, \ldots, X_{t-s'}, \ldots)$ and $(X_t, X_{t-1}, \ldots, X_{t-s'}, \ldots, X_{t-s}, \ldots)$ for any pair s, $s' < \infty$. Thus, for a finite k, we rewrite state space by the counts of N joint states,

$$\begin{aligned} C^t_{t-k} := \{&(C_1, C_2, \ldots, C_N): \\ &C_i = |\{x \in \{X_{t-k}, X_{t-k+1}, \ldots, X_t\} : x = i\}|\}. \end{aligned}$$

Over this counting state space C^t_{t-k}, we obtain the recursive equation on the stationary distribution

$$\begin{aligned} &P(C_1, C_2, \ldots, C_N) = \\ &\sum_{i=1}^{N} \pi_i\, P\big(C_1 - \delta_{i,1}, C_2 - \delta_{i,2}, \ldots, C_N - \delta_{i,N}\big). \end{aligned} \tag{7}$$

For $i = 1, \ldots, N$, π_i is the conditional probability

$$\pi_i = P\left(C_1, C_2, \ldots, C_N \,\middle|\, C_1 - \delta_{i,1}, C_2 - \delta_{i,2}, \ldots, C_N - \delta_{i,N}\right).$$

Given a probability of some initial state, we can compute the forward-in-time probabilities over these counting states, using equation (7) until k is sufficiently large. In the numerical implementation of this case, we removed counting states with a probability less than 10^{-10} for

https://doi.org/10.25088/ComplexSystems.24.1.49

computational efficiency. This rounding reduced the probability to less than 1% of the total probability 1 in our analysis.

2.4 Continuous-Time Limit Model

For consistency of notation, we write

$$R_{i,x}^{\alpha_i}\left(X_{t-k}^{t-1}\right) := \phi_{i,x}^{\alpha_i}\left(X_{t-k}^{t-1}\right) + \delta_{x,x_{t,i}}\, r_i(X_t)$$

and transform the learning parameters by

$$a_i = 1 - \alpha_i,\ \ b_i = \beta_i\,(1 - \alpha_i).$$

When $k \to \infty$, equation (2) can be rewritten as

$$P\left(x \,\middle|\, X_{-\infty}^{t}\right) = \frac{P\left(x \,\middle|\, X_{-\infty}^{t-1}\right) \exp\left(\beta_i\, \Delta \mathrm{R}_{i,x}^{\alpha_i}\left(X_{-\infty}^{t}\right)\right)}{\sum_{x=0}^{m-1} P\left(x \,\middle|\, X_{-\infty}^{t-1}\right) \exp\left(\beta_i\, \Delta \mathrm{R}_{i,x}^{\alpha_i}\left(X_{-\infty}^{t}\right)\right)},$$

where

$$\Delta \mathrm{R}_{i,x}^{\alpha_i}\left(X_{-\infty}^{t}\right) := R_{i,x}^{\alpha_i}\left(X_{-\infty}^{t}\right) - R_{i,x}^{\alpha_i}\left(X_{-\infty}^{t-1}\right).$$

Consider the limit when this update of probability measures takes place only after infinitely many interactions. This allows us to treat the given difference equation as a differential equation. With regard to the continuous-time limit, Sato and Crutchfield [9] derived the following ordinary differential equation of the marginal probability measure $P_i(y)$ for the i^{th} agent to take the action $y \in \mathcal{M}$. For integers $1 \le i \le n$,

$$\frac{\dot{P}_i(y)}{P_i(y)} = b_i\left(\bar{r}_i(y) - \sum_{x=1}^{m} P_i(x)\,\bar{r}_i(x)\right) + a_i \sum_{x=1}^{m} P_i(x) \log \frac{P_i(x)}{P_i(y)}, \tag{8}$$

where $\dot{y}$ denotes the derivative of y, and $\bar{r}_i(y)$ is the resulting reward that occurs when agent i chooses strategy y averaged over the other agents' strategy during the time interval between learning updates. This equation was solved numerically, and its trajectory and dynamical properties were analyzed in [9].

2.5 Consistency in the Different Approaches

Given Definition 1, each of the three approaches reviewed provides a different type of approximation to a learning-and-game model with infinite depth $k \to \infty$. Our interest here is whether these three models are consistent for $\alpha \approx 1$, with which Sato et al. found chaotic dynamics. The following technical reasons may affect whether the approaches are consistent.

https://doi.org/10.25088/ComplexSystems.24.1.49

The continuous-time limit model may provide a good approximation when its limiting condition holds sufficiently. To our understanding, this limit condition implicitly requires a learning-and-game model to have $k \to \infty$ and $a_i = 1 - \alpha_i \approx 0$ for every i, because it assumes the learning applies only after sufficiently long play. In fact, Sato et al. [7–9] have analyzed only special cases with $a_i < 0.03$. The applicability of this approximation remains to be analyzed.

The agent-based simulation provides an approximation up to sampling error whose magnitude depends on k and α_i. As noted before, for a large k and some $\alpha_i \approx 1$, the convergence of statistics using agent-based simulation may be quite slow.

The k^{th}-order Markov process gives an approximation up to a finite and very small k. As noted in Section 2.3.1, this limitation may be relaxed to some extent, except for the case $\alpha_i = 1$. The case $\alpha_i = 1$ is more tractable in computational terms, as described in Section 2.3.2. The stationary distribution in this special case is crucial for the analysis of consistency between the three models.

In the following section, we numerically test the consistency between these three models. Consistency will imply that the continuous-time limit model captures not just the special type of learning assumed in [7–9], but also the original properties of the reinforcement learning given by Definition 1. Otherwise, we need to further validate whether the agent-based simulation and the Markov process analysis are consistent with each other, in order to confirm their inconsistency with the continuous-time limit model.

3. The Continuous-Time Limit Model Revisited

In this section, we investigate the class of games, rock-paper-scissors, in which Sato and colleagues [7–9] have found generic dynamical classes including chaos. We analyze the rock-paper-scissors games between two agents using the continuous-time limit model and the other two methods described in the previous section. The rock-paper-scissors game [9] is defined as follows.

Definition 2 (2-agent-*k*-memory iterated rock-paper-scissors game). Using the notations in Definition 1, we set $n = 2$, $m = 3$, and

$$x_{t,i} := \begin{cases} 0 & \text{rock } (R) \\ 1 & \text{paper } (P) \\ 2 & \text{scissors } (S). \end{cases}$$

For integer $i = 1, 2$, we write the opponent $j = 3 - i$ for the i^{th} agent.

https://doi.org/10.25088/ComplexSystems.24.1.49

Let state X_t at each step t take one of the nine pairwise actions by the two agents and denote it by

$$X_t = (x_{t,i}, x_{t,j}) = \begin{cases} (0,0):\mathrm{RR}, & (0,1):\mathrm{RP}, & (0,2):\mathrm{RS}, \\ (1,0):\mathrm{PR}, & (1,1):\mathrm{PP}, & (1,2):\mathrm{PS}, \\ (2,0):\mathrm{SR}, & (2,1):\mathrm{SP}, & (2,2):\mathrm{SS}. \end{cases}$$

By taking an action $x_{t,i}$, the i^{th} agent receives the reward

$$r_i(x_{t,i}, x_{t,j}) := \begin{cases} 1, & \text{if } x_{t,j} - x_{t,i} \equiv 1 \pmod 3 \\ -1, & \text{if } x_{t,j} - x_{t,i} \equiv -1 \pmod 3 \\ \epsilon_i, & \text{if } x_{t,j} - x_{t,i} \equiv 0 \pmod 3. \end{cases}$$

We consider this game as $k \to \infty$, but its k^{th}-order Markov process analysis is performed for a finite k.

3.1 Results

First, as a sanity check, we replicated the numerical experiments in the exact settings studied in [9]. With the parameters $a = 0$, $b = 1$, $\epsilon_1 = -\epsilon_2 = 0.5$, we ran two simulations of the Hamiltonian class of the system with the initial conditions reported in [9]. Figure 1 shows the phase portraits (of Agent 1) of the quasiperiodic tori (left) and chaos (right), which respectively correspond to the two different initial conditions. We replicated these trajectories as reported in the previous study.

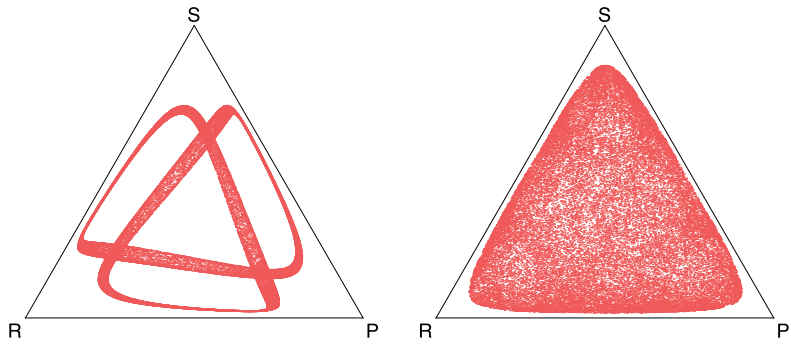

Figure 1. The (a) quasiperiodic tori and (b) chaos in the phase space of the continuous-time limit model with parameters $a = 0$, $b = 1$, $\epsilon_1 = -\epsilon_2 = 0.5$, and two different initial conditions.

https://doi.org/10.25088/ComplexSystems.24.1.49

As parameter b and initial values were not reported in [9], we replicated them with the parameters $a = 0.01$, $b = 1$, $\epsilon_1 = 0.5$, $\epsilon_2 = 0.025$, and several initial conditions drawn from random values. We typically obtained limit cycles as shown in Figure 2(a). With the same parameters excepting $\epsilon_2 = -0.365$, and several initial conditions drawn from random values, we typically obtained chaotic trajectories as shown in Figure 2(b). These results are quite similar to those reported in the previous study.

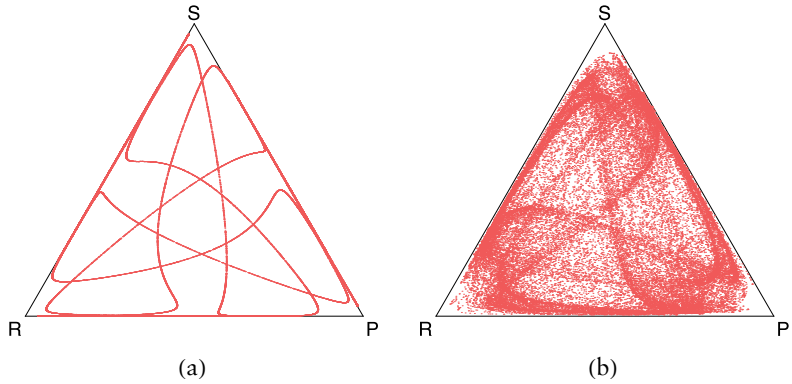

Figure 2. A limit cycle ((a): $\epsilon_2 = 0.025$) and chaotic trajectory ((b): $\epsilon_2 = -0.365$) in the phase space of the continuous-time limit model with parameters $a = 0.01$, $b = 1$, $\epsilon_1 = 0.5$, and random initial conditions.

Next, we compared the results obtained by the continuous-time limit model to those of the agent-based simulation and finite-state Markov process. Since it has the most representative outcome of the learning-and-game models presented by Sato and colleagues [9], we analyzed the game resulting in chaotic dynamics with parameters $0 < a \leq 1$, $b = 1$, $\epsilon_1 = 0.5$, $\epsilon_2 = -0.365$. We analyzed an average of 30 samples for each set of parameters with different initial conditions.

Using our best computational resources, we could reasonably compute the stationary distributions up to $k = 8$ for $a > 0$ and $k = 63$ for $a = 0$. As the continuous-time limit model does not share the same discrete time steps as the other class of models, the choice probabilities at each step are not comparable across the three models. Instead, we analyzed the marginal probability of outcomes of these models, which can be compared across the three models. Considering the symmetry of the three actions: rock, paper, and scissors, we only evaluated the probability of Agent 1's win, of Agent 1's loss, and of a draw. Figure 3(a) shows the probabilities of the win, loss, and draw as a function of a in the chaotic case (Figure 2(b)). The red, green, and blue

https://doi.org/10.25088/ComplexSystems.24.1.49

points show the sample probability of Agent 1's win, loss, and draw, respectively, in the agent-based simulations with 10^6 samples. The solid lines corresponding to the color show the probabilities calculated by the finite-state Markov process $k = 8$ ($a > 0$), and the corresponding triangles show those of $k = 63$ ($a = 0$). The results of the agent-based simulations (open circles) and the Markov process $k = 8$ (solid lines) had a high correlation coefficient (0.9716) and were tightly matched, except for the case $a < 0.1$. For $a \approx 1$, the Markov process of $a = 0$ gives a better fit to the result of the agent-based simulation.

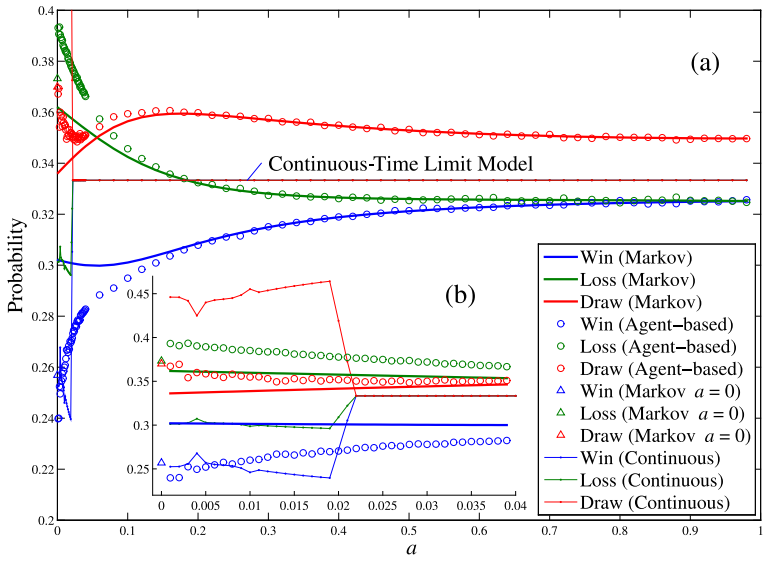

Figure 3. (a) Probabilities of a win, a loss by Agent 1, and a draw calculated by the agent-based simulation (circles), the Markov process with $k = 8$ (solid lines), the Markov process over the counting states $k = 63$, $a = 0$ (triangles), and the continuous-time limit model (lines with dots). (b) Those over the interval $a < 0.04$.

In contrast, the continuous-time limit model (the dashed lines in Figure 2(a)) had uniform probability 1 / 3 in all three cases, except for $a \leq 0.022$. For a closer look in Figure 2(b), the subpanel zooms in on the parameter interval $a < 0.04$. This subpanel shows that the probabilistic properties in the continuous-time limit model are quite different for $a \leq 0.022$. This bifurcation is found consistently in the previous study. With similar (but slightly different) parameters, Sato et al. [8, Figure 19] shows a bifurcation around $a \approx 0.02$. Even in this pa-

https://doi.org/10.25088/ComplexSystems.24.1.49

rameter interval, it is evident that the continuous-time limit model fits neither the agent-based nor the Markov process model. As a decreases, the probability of Agent 1's loss increases in both the agent-based simulation and the Markov process, but it becomes smaller than the baseline probability $1/3$ in the continuous-time limit model. The results of the Markov process in the special case $a = 0$ and $k = 63$ (triangles) are qualitatively consistent with those of the agent-based simulation and the Markov process with $k = 8$.

For further characterization of reinforcement learning, we analyzed the average rewards of the two agents in each approach (Figure 4). This additional analysis demonstrated that the average reward for each agent in the agent-based simulation approached 0.0238 (black horizontal line in Figure 4). This value is nearly equal to $(\epsilon_1 + \epsilon_2)/6 = 0.0225$, which is the equal share of rewards for a uniform random play in this game. The results of the Markov processes ($k = 8$ and $k = 63$) were similar to that of the agent-based simulation, which showed the general trend of balancing the average

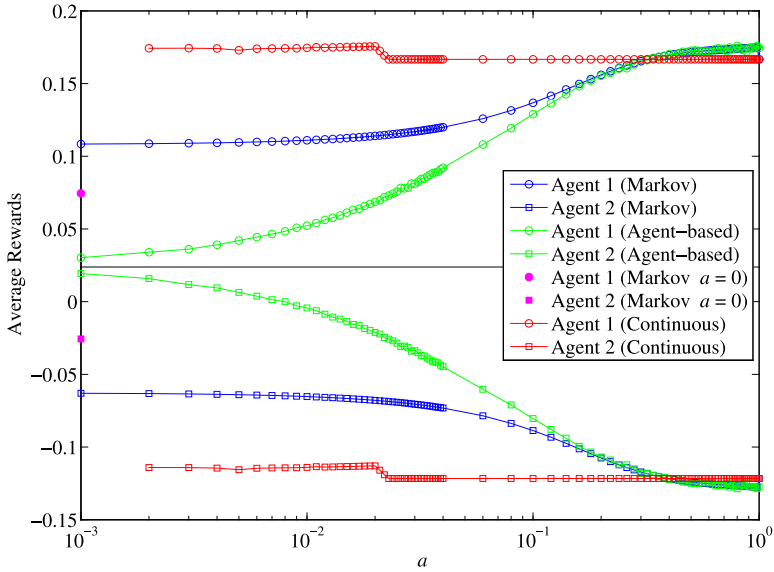

Figure 4. Average rewards for Agent 1 (circle) and Agent 2 (square) in the Markov process analysis $k = 8$ (blue), the agent-based simulation (green), and the continuous-time limit model (red) as a function of a. Those of the Markov process over the counting state space with $k = 63$, $a = 0$ are also shown (purple). The black horizontal line shows the average reward 0.0238, which is the average of the two agents' rewards in the agent-based simulation approach.

https://doi.org/10.25088/ComplexSystems.24.1.49

rewards between the two agents as $a \to 0$ and k increases. This result suggests that two agents with the same learning parameters $a_1 = a_2$ and $b_1 = b_2$ would have equivalent average rewards in the limit of $a_1 = a_2 \to 0$, although the agents have asymmetric rewards for their actions ($\epsilon_1 > \epsilon_2$). In contrast, the average rewards in the continuous-time limit model did not match the trend toward the balanced point (Figure 4). As the balancing in the average rewards characterizes reinforcement learning, this result suggests that the continuous-time limit model does not capture this essential property of reinforcement learning.

These results show that the continuous-time limit model is generally inconsistent with the other two models, while the agent-based simulation and Markov process show qualitative consistency over broad parameter intervals.

3.2 Discussion

The inconsistency of the outcome of the continuous-time limit model with the other two models suggests that the continuous-time model reflects neither the quantitative nor the qualitative nature of reinforcement learning. This inconsistency sheds doubt on the theoretical claim by Sato and colleagues [7–9] that reinforcement learning can lead to chaotic dynamics, as their limit model does not approximate reinforcement learning. It is likely that the continuous-time limit approximation leads to another type of "learning" that has little to do with reinforcement learning. Thus, in contrast to the conclusions of Sato and colleagues in [7–9], we warn that there is no support for the continuous-time limit model in relation to reinforcement learning leading to generic dynamical classes in a game.

4. Consistency Analysis of Agent-Based Simulation and the Markov Process

Given the results of Section 3, the remaining issue is a more systematic test regarding the consistency between agent-based simulation and its corresponding Markov process analysis. Although the two results show reasonable fits in the analysis on rock-paper-scissors, they show some dissociation as $a \to 0$. The following analysis numerically investigates whether this dissociation converges to zero asymptotically as $k \to \infty$.

Since the rock-paper-scissors game has a relatively large combination $N = 9$ at each step, we could only reasonably calculate up to $k = 8$. We employ a game with a smaller combinatorial size in order

https://doi.org/10.25088/ComplexSystems.24.1.49

to analyze the consistency between the two models for a relatively large k. Specifically, we analyzed an iterated prisoner's dilemma game that has the minimal number of states $N = 4$. We consider two agents playing the iterated games by choosing an action at each step using reinforcement learning. We suppose that the consistency between agent-based simulation and Markov process analysis would be essentially common across multiple learning-and-game models. Thus, the choice of the prisoner's dilemma purely stems from its minimal combinatorial space.

The prisoner's dilemma is a classic game that has been studied as a minimal model of tradeoff between cooperation and defection. The basic game has been extended to games with multi-agent, iterated steps, stochastic strategy, situation under noise, and a certain topology of agents' interactions [11, 14, 15]. In the iterative variant of models, each agent can adaptively choose its action on the basis of a series of past actions. One of the simplest cases is completely analyzed based on the finite Markov formalism [12], but more general cases remain for further research.

The iterated prisoner's dilemma as a learning-and-game model is defined as follows.

Definition 3 (2-agent-*k*-memory iterated prisoner's dilemma). In the notations defined in Section 2, we set $n = 2$, $m = 2$, and

$$x_{t,i} := \begin{cases} 0, & \text{cooperation} \\ 1, & \text{defection.} \end{cases}$$

For each step t and agent 1 and 2, let us write the set of the four states

$$X_t := 2\, x_{t,1} + x_{t,2} + 1.$$

For integer $i = 1, 2$, the i^{th} agent with its opponent $j = 3 - i$ receives the reward

$$r_i\left(x_{t,i}, x_{t,j}\right) := R_{x_{t,i},x_{t,j}}$$

by taking an action $x_{t,i}$. The rewards satisfy $R_{01} < R_{11} < R_{00} < R_{10}$ and $(R_{01} + R_{10}) < 2\, R_{00}$ in order for this game to be classified as a prisoner's dilemma.

4.1 Simulation and Analysis

In the agent-based simulation, we initialized $Q_0 = 0$, and each action of each agent is randomly chosen using equation (3) in Definition 1. We discarded the first 1000 samples as transients and used the subsequent 3×10^6 or 3×10^5 samples to compute the stationary distribution for each joint state X_t^{t+k} for a given k by treating it as a k^{th}-order

https://doi.org/10.25088/ComplexSystems.24.1.49

Markov process. In the Markov process analysis, for each $k \leq 9$ we used the transition matrix in equation (5) to compute the stationary distribution over the states X_t^{t+k}. We analyzed the iterated prisoner's dilemma with the rewards $R_{00} = 1$, $R_{01} = -2$, $R_{10} = 2$, and $R_{11} = 0$.

4.2 Error Analysis

For $\alpha_1 = \alpha_2 = 0.5$, $\beta_1 = \beta_2 = 0.5$, and $k \leq 9$, Figure 5 shows the sample probabilities of N^k states in the agent-based simulation as a function of the stationary probabilities in the corresponding k^{th}-order Markov process. Figure 6 shows another case with the parameters $\alpha_1 = \alpha_2 = 0.7$ and $\beta_1 = \beta_2 = 1$. The results show a good fit between the two methods as k increases.

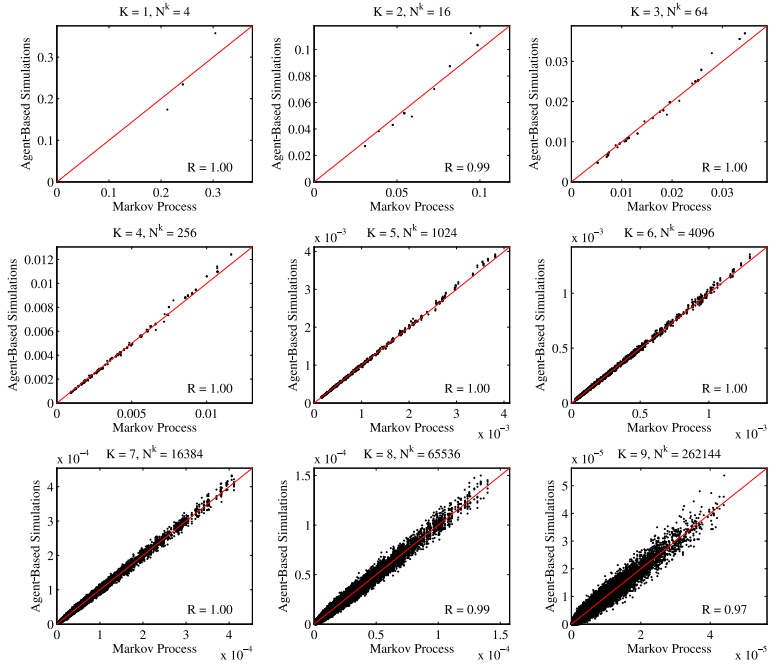

Figure 5. The stationary probabilities of N^k states in the agent-based simulations (y axis) as a function of those of the k^{th}-order Markov process (x axis) for $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 0.5$ and $k \leq 9$.

https://doi.org/10.25088/ComplexSystems.24.1.49

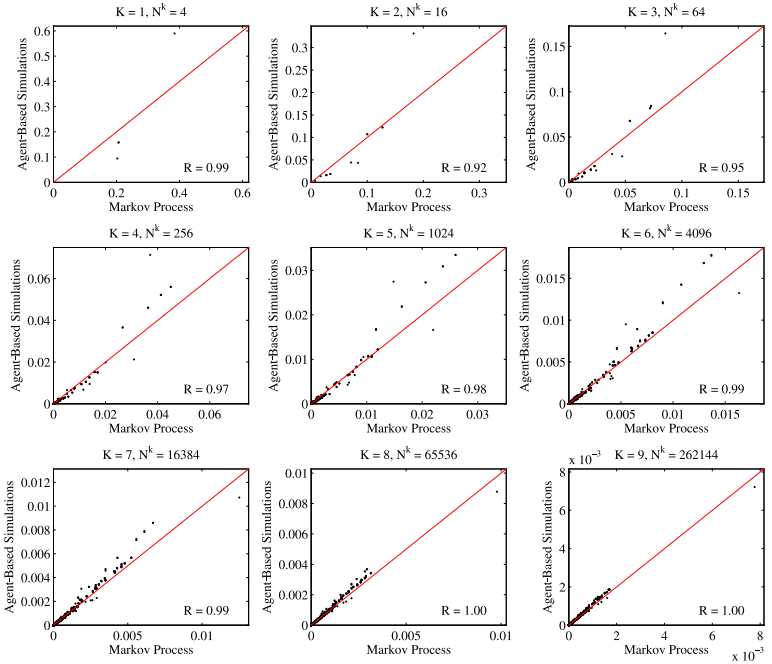

Figure 6. The stationary probabilities of N^k states in the agent-based simulations (y axis) as a function of those of the k^{th}-order Markov process (x axis) for $\alpha_1 = \alpha_2 = 0.7$, $\beta_1 = \beta_2 = 1$, and $k \leq 9$.

For a more rigorous statistical test on model fitting, we evaluated the chi-squared statistics [16]

$$\chi^2\left(N^k - 1\right) = \sum_{i=1}^{N^k} \frac{(n_i - n\,p_i)^2}{n\,p_i} \tag{9}$$

where, for joint states $1 \leq i \leq N^k$, $n = \sum_{i=1}^{N^k} n_i$, p_i is the stationary probability in the k^{th}-order Markov process, and n_i is the count of joint state i in an agent-based simulation. If each state in the agent-based simulation is sampled from the distribution $\left(p_1, p_2, \ldots, p_{N^k}\right)$, $\chi^2\left(N^k - 1\right)$ follows the chi-squared distribution χ_{N^k-1} with $N^k - 1$ degrees of freedom. We set the upper bound of the sampling error $\hat{\chi}$ such that the cumulative distribution $\chi_{N^k-1}\left(\chi^2\left(N^k - 1\right) \leq \hat{\chi}\right) = 0.975$. We say that a Markov process fits the corresponding agent-

https://doi.org/10.25088/ComplexSystems.24.1.49

based simulation if its chi-squared statistics $\chi^2\left(N^k - 1\right)$ are smaller than the supposed bound $\hat{\chi}$.

Figure 7 shows $\chi^2\left(N^{\min(k,5)} - 1\right)$ of the 15 cases with the parameters $\alpha := \alpha_1 = \alpha_2 = 0.3, 0.5, 0.7, 0.9, 1$; $\beta_1 = \beta_2 = 0.1, 0.5, 1$; and $k = 1, 2, \ldots, 12$ for $\alpha < 1$ and $1 \leq k \leq 500$ for $\alpha = 1$. With our computational resource, $k = 13$ for $\alpha < 1$ was the maximum practically computable in a reasonable time, due to the exponential growth of the state space ($N^{13} \approx 10^{7.82}$). Due to the special nature of $\alpha = 1$, we were able to compute over $k = 500$ for it and found the stationary distribution was converging for $k > 1000$, as far as its marginal distribution is concerned.

The red line in each panel shows the upper bound $\hat{\chi}$ of sampling error of the agent-based simulation with 3×10^6 for $\alpha < 1$ and with 3×10^5 for $\alpha = 1$. In general, the larger state space gives a stricter statistical test on the model fit but requires a larger sample size for the agent-based simulation. Considering the tradeoff between computational cost and statistical power, we chose to evaluate the chi-squared errors in marginal distributions in the state space $N^5 = 1024$ for $k \geq 5$ and $\alpha < 1$. For $k > 5$ and $\alpha < 1$, each of the N^k states $X_{t-k+1}^t = (X_t, \ldots, X_{t-k+1})$ is mapped to its corresponding state $X_{t-4}^t = (X_t, \ldots, X_{t-4})$, and the chi-squared values of these marginal probabilities over N^5 states were analyzed. For $\alpha = 1$, the stationary distribution of every state with probability larger than 10^{-10} was analyzed. The cases with deviations smaller than the upper bound are shown in the filled circles, and all other cases are represented by open circles.

These results show a general trend that the Markov processes fit to the corresponding agent-based simulations with a sufficiently large k. For small α and β, a small k is enough for a good fit, and for large α and β, a relatively larger k is needed. As $\alpha \to 1$ and $\beta \to \infty$, the agents learn with an infinite history length, and the fitting of a finite Markov process becomes worse. In fact, among the cases we analyzed, the case with the larger parameters $0.7 \leq \alpha < 1$ did not show any well-fitting cases within this range of $k \leq 10$ (Figure 7). However, for $\alpha = 1$, the finite Markov process showed notably smaller errors than the theoretical error bound for every $1 \leq k \leq 500$, excepting a few cases (Figure 7). This confirms that the Markov process analyses fit the corresponding agent-based simulations over a broader range $\alpha < 0.7$ or $\alpha = 1$ with sufficiently large k.

https://doi.org/10.25088/ComplexSystems.24.1.49

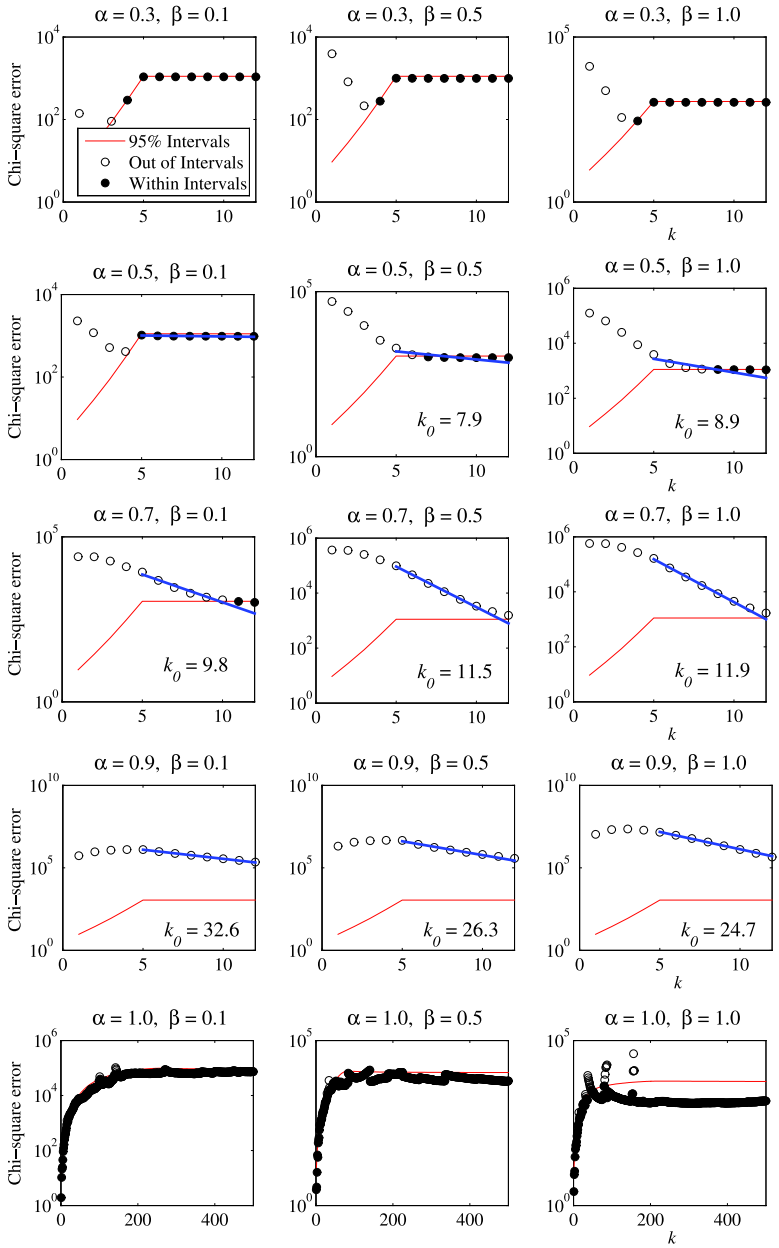

Figure 7. The chi-squared statistics $\chi^2\left(N^k - 1\right)$ for the sampling errors in the agent-based simulations from the corresponding Markov process.

https://doi.org/10.25088/ComplexSystems.24.1.49

As discussed in Section 2.3.1, the squared errors between the k^{th}-order Markov process and the true model reduce exponentially as a function of k. Our mathematical analysis discussed in Section 2.3.1 states that exponential decays of errors are generally slower for larger α. Testing this statement, we analyzed the slopes of the expected error shown as the blue lines in Figure 7. As expected, their exponential decays are slower as α increases. These results confirmed the theoretical properties of the Markov process in Section 2.3.1 and Appendix A. Although it is not directly computable with a large k, this exponential reduction of errors suggests a practical consistency between the Markov process and agent-based simulation for a broad interval of $\alpha < 1$.

We estimated the expected minimal k_0 (described in each panel in Figure 7) in which the extrapolated error line (blue) meets the sampling error upper bound (red) for each set of parameters. For $\alpha \approx 1$, this suggests that the necessary k for the Markov process to substantially approximate the model as $k \to \infty$ is quite large and not easily computable. Nonetheless, this error analysis with an extrapolated line would be useful when we wish to evaluate a sufficient sample size for the convergence of an agent-based simulation. The Markov process analysis is useful for a smaller $\alpha < 0.7$, since it does not have any sampling error. With $\alpha > 0.7$, an agent-based simulation or the Markov process analysis with $\alpha = 1$ may be more reliable.

5. General Discussion

In this study, we investigated a game with reinforcement learning using three different approaches and numerically investigated whether reinforcement learning leads to generic dynamical classes in the game. The continuous-time limit model proposed in [7–9] has been thought to show the theoretical relationship between reinforcement learning and complex dynamics in a game. Since this theoretical claim is limited for special learning after infinitely many actions, we tested whether the continuous-time limit model can be continuously connected to the other models without this assumption.

Our analysis showed that the outcomes of agent-based simulation and the Markov process analysis are consistent. Our additional analysis showed that the average rewards of the two agents tended to converge to a balanced point in both the Markov process analysis and agent-based simulation. This finding suggests that two reinforcement learners with equal learning parameters approach an equilibrium with sufficiently large k and small a. However, this analysis also revealed

https://doi.org/10.25088/ComplexSystems.24.1.49

that the continuous-time limit model did not exhibit a similar trend toward an equilibrium. In sum, the comparison with these two other approaches revealed that the continuous-time limit model does not capture the basic qualitative outcomes of a game with reinforcement learning, such as who wins against whom or the equilibrium in the average rewards. Importantly, the Markov process analysis with $a = 0$ showed a converging stationary distribution, which means this game has a unique fixed point to which it converges (and thus, the game cannot be a chaotic attractor). Thus, this analysis does not support Sato and Crutchfield's [9] claim that reinforcement learning leads to complex dynamics such as chaos in a game. To put it another way, our analysis suggests that two reinforcement learners in these games cannot be universal, as this class of computation cannot encode a complex class of phenomena sufficiently [10].

The second study on the iterated prisoner's dilemma gave an extended investigation confirming the agreement between the agent-based simulation and corresponding Markov process for relatively large k.

We conclude the paper by summarizing two benefits of using the finite-state Markov approach. First, it offers another computational method for agent-based simulations and can be used for a sanity check of these kinds of models, as it has no sampling error, unlike agent-based simulation. Second, unlike in agent-based simulation, we can characterize the potential dynamics directly from the mathematical property of a given transition matrix. A drawback of using the Markov process analysis is its computational cost for a large state space. For this issue, we offer a potential collaboration with the agent-based simulation of a game and the corresponding Markov process analysis. This hybrid approach can be a potentially powerful tool to explore a broad class of games with multiple agents.

Acknowledgments

This study was supported by the NeuroCreative Lab, Grant-in-Aid for Scientific Research B No. 23300099, and Grant-in-Aid for Exploratory Research No. 25560297.

Appendix

A. Errors in a Finite-State Markov Process

For a model with $k \to \infty$ and $\alpha < 1$ formulated in Definition 1, we evaluate the errors of a Markov process with a finite k defined in Sec-

https://doi.org/10.25088/ComplexSystems.24.1.49

tion 2.3. Let $\theta \in \mathbb{R}^{N^k}$ and $\overline{\theta} \in \mathbb{R}^{N^k}$, where $\theta^T \theta = \overline{\theta}^T \overline{\theta} = 1$, be the eigenvector corresponding to the largest eigenvalue $\lambda = 1$ of the transition matrix $Q \in \mathbb{R}^{N^k \times N^k}$ and $Q^T \in \mathbb{R}^{N^k \times N^k}$. Suppose the true k^{th}-order transition matrix is Q_0 and its stationary vector is θ_0 in the limit $k \to \infty$. The sum of squared errors is defined by

$$\epsilon^2 = (\theta - \theta_0)^T (\theta - \theta_0).$$

When each cell in the difference matrix $\Delta = Q - Q_0$ is sufficiently smaller, the difference is approximated by the Taylor series up to the first order

$$\theta_0 \approx \theta - \left(\text{vec}\,(\Delta)^T \frac{\partial \theta}{\partial \text{vec}\,(Q)} \right)^T,$$

where we denote the vectorization operator to a matrix $X \in \mathbb{R}^{M \times N}$ by

$$\text{vec} : X \to \left(X_{1,1}, \ldots, X_{M,1}, \ldots, X_{1,N}, \ldots, X_{M,N}\right)^T.$$

Then we find

$$\epsilon^2 \approx \text{vec}\,(\Delta)^T \frac{\partial \theta}{\partial \text{vec}\,(Q)} \left(\frac{\partial \theta}{\partial \text{vec}\,(Q)} \right)^T \text{vec}\,(\Delta).$$

The partial differential of θ with respect to the vector vec (Q) is

$$\frac{\partial \theta}{\partial \text{vec}\,(Q)} = \left(\theta \otimes E_{N^k}\right) \left(E_{N^k} - \left(\theta^T \overline{\theta}\right)^{-1} \overline{\theta}\, \theta^T \right) \left(\lambda\, E_{N^k} - Q^T\right)^{\dagger},$$

where $E_N \in \mathbb{R}^{N \times N}$ is the $N \times N$ unit matrix, $\otimes$ denotes the Kronecker product, and $X^{\dagger} = X^T \left(X\, X^T\right)^{-1}$ denotes the Moore–Penrose generalized inverse matrix of X. For most cases of the learning-and-game model, Q has its second-largest eigenvalue $|\lambda_2| \ll \lambda$, and $Q \approx \theta \overline{\theta}^T$. This gives

$$\left(E_{N^k} - \left(\theta^T \overline{\theta}\right)^{-1} \overline{\theta}\, \theta^T \right) \left(\lambda\, E_{N^k} - Q^T\right)^{\dagger} \approx E_{N^k},$$

and the sum of squared errors ϵ^2 is approximately a quadratic function of Δ:

$$\epsilon^2 \approx (\Delta\theta)^T\, \Delta\theta.$$

With $\alpha < 1$, the difference Δ decreases exponentially as a function of k in equation (3). Therefore, the sum of squared errors ϵ^2 decreases exponentially as k increases. The exponential decays of the weighted

https://doi.org/10.25088/ComplexSystems.24.1.49

sum of squared errors are numerically demonstrated by the results shown in Figure 7.

B. Efficient Computation of Stationary Distribution

Here we show an efficient algorithm to compute the stationary distribution θ for an arbitrary transition matrix Q of the N-state k^{th}-order Markov process in equation (5).

Before showing the main result, let us introduce additional notations. Let us define the inverse of indexing map: For $1 < j \leq k-1$,

$$g_N(i, j) := 1 + \mathrm{mod}_N\left(\left\lfloor \frac{i-1}{N^{j-1}} \right\rfloor\right),$$

where $\mathrm{mod}_N(x) := x \bmod N$ and $y = \lfloor x \rfloor$ is the maximal integer $y \leq x$. Then,

$$i = h_{N,k}((g_N(i, k), g_N(i, k-1), \ldots, g_N(i, 1))).$$

The transition matrix Q defined in equation (5) can have at most N positive elements in each column. For the j^{th} column and $1 \leq i \leq N$, the nonzero element is

$$q_{i,j} := Q_{f_{N,k}(i,j),j},$$

where

$$f_{N,k}(i, j) := h_{N,k}((i, g_N(j, k), g_N(j, k-1), \ldots, g_N(j, 2))).$$

Let us denote the unit matrix by $E_N \in \mathbb{R}^{N \times N}$ and the unit vector

$$e_{N,i} := \left(0, \ldots, 0, \overset{i}{\overset{\vee}{1}}, 0, \ldots, 0\right)^T \in \mathbb{R}^N.$$

We define a special permutation matrix called a *commutation matrix* [17] by

$$C_{M,N} := \left(E_M \otimes e_{N,1}, E_M \otimes e_{N,2}, \ldots, E_M \otimes e_{N,N}\right).$$

By the commutation matrix, an arbitrary $M \times N$ matrix X holds

$$\mathrm{vec}(X) = C_{M,N}\,\mathrm{vec}\left(X^T\right).$$

Now we are ready to state the main result.

https://doi.org/10.25088/ComplexSystems.24.1.49

Theorem 1. Let $Q \in \mathbb{R}^{N^k \times N^k}$ be a transition matrix defined in equation (5). Then, it has the following decomposition:

$$Q = C_{N,N^{k-1}}\, \overline{Q},$$

where

$$\overline{Q} = \begin{pmatrix} Q_1 & & & \\ & Q_2 & & 0 \\ & & \ddots & \\ 0 & & & Q_{N^{k-1}} \end{pmatrix}$$

with the block diagonal matrix Q_m in which the (i, j) element

$$\{Q_m\}_{i,j} = Q_{a,b}$$

where

$$a = f_{N,k}\,(i, N\,(m-1)+j),\ \ b = N\,(m-1)+j.$$

Proof. We can write the transition matrix

$$Q = \sum_{i=1}^{N^{k-1}} e^T_{N^{k-1},i} \otimes Q_i \otimes e_{N^{k-1},i}.$$

Similarly,

$$\overline{Q} = \sum_{i=1}^{N^{k-1}} e^T_{N^{k-1},i} \otimes e_{N^{k-1},i} \otimes Q_i,$$

and

$$C_{N,N^{k-1}} = \sum_{i=1}^{N^{k-1}} e^T_{N^{k-1},i} \otimes E_N \otimes e_{N^{k-1},i}.$$

Then it is easy to see

$$Q = C_{N,N^{k-1}}\, \overline{Q}. \square$$

With this theorem, we obtain the following result.

Corollary 1. For an arbitrary vector $x \in \mathbb{R}^{N^k}$ and transition matrix $Q \in \mathbb{R}^{N^k \times N^k}$ with its block diagonal matrices $Q_i \in \mathbb{R}^{N \times N}$; $i = 1, \ldots, N^{k-1}$,

$$Q\,x = \mathrm{vec}\left(\left(Q_1\,x_1, Q_2\,x_2, \ldots, Q_{N^{k-1}}\,x_{N^{k-1}}\right)^T\right), \tag{B.1}$$

https://doi.org/10.25088/ComplexSystems.24.1.49

where $x_i \in \mathbb{R}^N \left(i = 1, \ldots, N^{k-1}\right)$ satisfies

$$x = \mathrm{vec}\left(\left(x_1, x_2, \ldots, x_{N^{k-1}}\right)\right).$$

The left-hand side of equation (B.1) is needed in numerically solving the eigenvalue problem, and the multiplication costs computational complexity $O\left(N^3{}^k\right)$ in general. The computational complexity of the matrix multiplication at the right-hand side in equation (B.1) is $O\left(N^{k+1}\right)$. Although the computational complexity in both forms is still an exponential function of k, the latter gives relatively efficient computation. Exploitation of this mathematical property relaxes the computational problem of a Markov process analysis to some extent.

References

[1] J. F. Nash, Jr., "Equilibrium Points in *n*-Person Games," *Proceedings of the National Academy of Sciences of the United States of America*, **36**(1), 1950 pp. 48–49. doi:10.1073/pnas.36.1.48.

[2] C. Camerer and T. H. Ho, "Experience-Weighted Attraction Learning in Normal Form Games," *Econometrica*, **67**(4), 1999 pp. 827–874. doi:10.1111/1468-0262.00054.

[3] C. F. Camerer, "Behavioural Studies of Strategic Thinking in Games," *Trends in Cognitive Sciences*, **7**(5), 2003 pp. 225–231.

[4] A. E. Roth and I. Erev, "Learning in Extensive-Form Games: Experimental Data and Simple Dynamic Models in the Intermediate Term," *Games and Economic Behavior*, **8**(1), 1995 pp. 164–212. doi:10.1016/S0899-8256(05)80020-X.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press, 1998.

[6] T. Börgers and R. Sarin, "Learning through Reinforcement and Replicator Dynamics," *Journal of Economic Theory*, **77**(1), 1997 pp. 1–14. doi:10.1006/jeth.1997.2319.

[7] Y. Sato, E. Akiyama, and J. D. Farmer, "Chaos in Learning a Simple Two-Person Game," *Proceedings of the National Academy of Sciences of the United States of America*, **99**(7), 2002 pp. 4748–4751. doi:10.1073/pnas.032086299.

[8] Y. Sato, E. Akiyama, and J. P. Crutchfield, "Stability and Diversity in Collective Adaptation," *Physica D: Nonlinear Phenomena*, **210**(1–2), 2005 pp. 21–57. doi:10.1016/j.physd.2005.06.031.

[9] Y. Sato and J. P. Crutchfield, "Coupled Replicator Equations for the Dynamics of Learning in Multiagent Systems," *Physical Review E*, **67**, 2003 p. 015206. doi:10.1103/PhysRevE.67.015206.

https://doi.org/10.25088/ComplexSystems.24.1.49

[10] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[11] T. W. Sandholm and R. H. Crites, "Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma," *Biosystems*, **37**(1–2), 1996 pp. 147–166. doi:10.1016/0303-2647(95)01551-5.

[12] M. Nowak, "Stochastic Strategies in the Prisoner's Dilemma," *Theoretical Population Biology*, **38**(1), 1990 pp. 93–112. doi:10.1016/0040-5809(90)90005-G.

[13] E. Seneta, *Non-negative Matrices and Markov Chains*, New York: Springer, 2006.

[14] R. Axelrod, *The Evolution of Cooperation*, New York: Basic Books, 1984.

[15] T. Galla, "Intrinsic Noise in Game Dynamical Learning," *Physical Review Letters*, **103**(19), 2009 p. 198702. doi:10.1103/PhysRevLett.103.198702.

[16] K. Pearson, "On the Criterion That a Given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such That It Can Be Reasonably Supposed to Have Arisen from Random Sampling," *Philosophical Magazine Series 6*, **50**(302), 1900 pp. 157–175. doi:10.1080/14786440009463897.

[17] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, rev. ed., New York: John Wiley, 1999.

https://doi.org/10.25088/ComplexSystems.24.1.49