Emergence of Chaotic Time Series by Adversarial Imitation Learning

Seiya Yamazaki, Hiroyuki Iizuka and Masahito Yamamoto

Graduate School of Information Science and Technology, Hokkaido University
yamazaki@complex.ist.hokudai.ac.jp

Abstract

Bird song is one of the phenomena that increase in complexity through evolution. A complex song is known to be advantageous for survivability and birds are known to learn how to sing a song from each other. From these facts, we have a hypothesis that adversarial imitation learning plays a major role in the evolution process of a complex song. There is a previous study that demonstrates the complexation of a bird song time series by modeling the process of adversarial imitation learning using a logistic map. However, the real bird songs have much variety and time dependencies, like grammar. Therefore, in this study, adversarial imitation learning is modeled using an artificial neural network that can approximate any function. The network learns adversarial imitation using the gradient descent method. By making such changes, the results of our study show that the generated bird songs evolve through the process of adversarial imitation learning to chaos, as seen in the previous models.

Introduction

Chirping of birds, represented by a Bengalese finch or nightingale, sounds like a song because the time series of phonemes is not ordered randomly but has patterns and grammars. It is known that birds which can generate more complex songs have an advantage in survivability. The songs are used for territorial claims (Catchpole, 1981) and courtship of females (ten Cate and Okanoya, 2012). Ten Cate and Okanoya also show that the Bengalese finches gradually change their own song while imitating each other.

Given that there is an advantage that complex songs present in the sexual selection process of the Bengalese finches and when they learn how to sing a song from the others, we have a hypothesis that adversarial imitation learning plays a major role in the evolution of complex songs. Adversarial imitation learning involves individuals imitating others' songs while their song should not be imitated by the others. To imitate the others' song, the generation models of the songs need to be similar: but if these models are too similar, it implies that the others can imitate as well. We postulate that this kind of contradictory learning pressure produces complex songs.

Suzuki and Kaneko showed that bird song time series becomes more complex by modeling the adversarial imitation learning of birds (Suzuki and Kaneko, 1994) and that the evolution of a bird song is directed to the edge of chaos. Since a bird song can be modeled as time series data, the logistic map is used to generate a bird song as a time series. These researchers showed that the time series of the bird songs become more complex while imitating each other under the fitness function that gives better rewards to individuals that can imitate others but are not imitated by the others. Because the aim of their study is to show the pathways of evolution to the edge of chaos, the song generation model is limited to the logistic map and evolution of the parameter of the nonlinearity of the map. However, the real bird songs have much variety and time dependencies, similar to that of grammar.

In this study, to realize such a variety in the simulation, the logistic map is replaced with an artificial neural network as the generation model for bird songs. The neural network is updated to learn and imitate the other songs and not to be imitated by the others at the same time. In such a situation, we investigate whether the generated bird songs evolve to chaos, as in the previous models, and what kind of situation makes the bird songs chaos. The different social relationships will be tested.

Simulation Model

Adversarial imitation learning

Suzuki and Kaneko showed that the time series evolves to a chaotic state due to the fitness function based on the adversarial imitation learning (Suzuki and Kaneko, 1994). As the generation model for a time series, the logistic map is used. In this paper, adversarial imitation learning is modeled with an artificial neural network and the gradient descent method.

Our simulation model consists of a time series generation phase as its own original bird songs and imitation phase to imitate the other bird songs. The time series generation is performed by a feed-forward neural network. The artificial neural network is updated based on adversarial imitation learning. Our model is explained using a case that

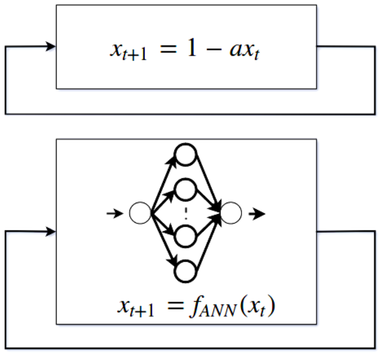

Figure 1: Top: Generation by the logistic map. Bottom: Generation by ANN.

involves two individuals interacting with and imitating each other, nonetheless, it can be easily extended to a scenario with more number of individuals.

Generation of their own bird songs

The bird songs are generated by an artificial neural network. When birds sing a song as their own bird song, they generate the time series of songs without listening to the other. Suzuki and Kaneko used the logistic map as the time series generation model and the songs were generated by the following equation (Suzuki and Kaneko, 1994),

$$x_{t+1} = 1 - ax_t^2, \tag{1}$$

where x_t is a single phoneme at time t and a is a nonlinearity parameter. During the evolution, only parameter a evolves in their model. Depending on the value of this parameter, the dynamics become periodic or chaotic.

In our model, the logistic map is replaced with an artificial neural network that can approximate any function that has a sufficient number of nodes and appropriate weights. The neural network that has a single input and output can generate the time series of the bird songs while listening to the individuals' own songs as follows:

$$x_{t+1} = f_{ANN}(x_t), \tag{2}$$

$$= g(\sum_{i=1}^{N} w_i(g(v_i x_t + b_i^1)) + b^2), \tag{3}$$

where w_i and v_i are weight, b_i^1 and b^2 are bias, N indicates the number of hidden neurons, and $g(x)$ is the activation function. The tanh function is used here. The time series $(x_1, x_2, \ldots, x_T)$ are generated by receiving their own outputs repeatedly.

Imitation of the other songs

When an individual bird imitates the other birds' songs, it generates the songs while listening to the other songs. The schematic view of the imitation process is shown in Fig. 2.

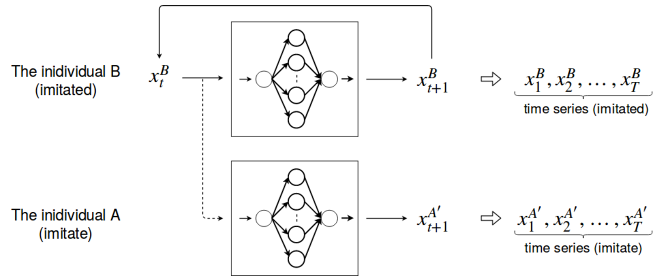

Figure 2: Individual B generates its own time series while listening to its own song, but individual A generates its own time series while listening to B's songs.

Let us explain the imitation phase in the case where individual A imitates B. Individual B generates its own bird song $\{x_1^B,\ x_2^B,\ \ldots,\ x_T^B\}$ using the neural network as follows,

$$x_{t+1}^B = f_{ANN}^B(x_t^B). \tag{4}$$

Individual A generates the time series $\{x_1^{A'},\ x_2^{A'},\ \ldots,\ x_T^{A'}\}$ using the neural network f_{ANN}^A while listening to the songs of individual B, that is, receiving the time series of individual B as follows:

$$x_{t+1}^{A'} = f_{ANN}^A(x_t^B). \tag{5}$$

When individual B imitates A, the roles are exchanged. B generates the bird songs while listening to the songs generated by A. These interactions are expressed as follows,

$$x_{t+1}^A = f_{ANN}^A(x_t^A), \tag{6}$$

$$x_{t+1}^{B'} = f_{ANN}^B(x_t^A). \tag{7}$$

Each individual has its own specific initial values to start the bird songs. As well as the weights of the network, the initial values are also updated by learning. To restrict the initial values in $[-1, 1]$, x_0 is obtained through the tanh function, $x_0 = \tanh(x_0^{'})$. The same initial values are used to generate their songs for both generation and imitation phases.

Objective function

The time series generated in the generation and imitation phases are evaluated in terms of the adversarial imitation learning. The individual is evaluated by how well it has imitated and how much it has not been imitated. Two types of errors are introduced to facilitate such an evaluation. One is the imitating-loss and the other is the imitated-loss. Let us explain these based on the case where individual A imitates B. The imitating-loss of A indicates how close the time series imitated by A is to B's song and is calculated by eq (8). The imitated-loss of A indicates how far the original song

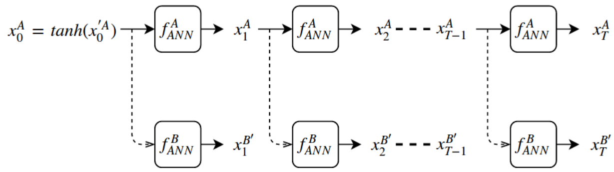

Figure 3: Individual A generates its own time series using its own output, but individual B generates its own time series using A's input. Therefore, the initial value of A can be applied using back-propagation through time (BPTT).

of A is from the songs imitated by B. The equation can be expressed by eq (9) as a minimization problem.

$$E^A_{imitating} = \frac{1}{T}\sum_{t=1}^{T}(x_t^B - x_t^{A'})^2, \tag{8}$$

$$E^A_{imitated} = \frac{1}{T}\sum_{t=1}^{T}(2 - |x_t^A - x_t^{B'}|)^2. \tag{9}$$

Imitating-loss and imitated-loss of B are also calculated in the same manner, as shown eq (10) and (11). Thus, the revised equations are as follows:

$$E^B_{imitating} = \frac{1}{T}\sum_{t=1}^{T}(x_t^A - x_t^{B'})^2, \tag{10}$$

$$E^B_{imitated} = \frac{1}{T}\sum_{t=1}^{T}(2 - |x_t^B - x_t^{A'}|)^2. \tag{11}$$

Imitated-loss is also used for updating the initial value of $x_0^{'}$. To learn the initial value, the errors are back-propagated to the initial value by considering the generating process of songs as the recurrent neural network like $x_t = f_{ANN}(f_{ANN}(\dots(f_{ANN}(\tanh(x_0^{'})))))$. Therefore, the initial value can be updated by BPTT (Robinson and Fallside, 1987; Chauvin and Rumelhart, 1995) such that each value of the two time series are more distant.

Those errors are integrated to the total errors of the individual, which can be expressed as follows:

$$E^A = E^A_{imitating} + E^A_{imitated}, \tag{12}$$

$$E^B = E^B_{imitating} + E^B_{imitated}. \tag{13}$$

The weights and biases of both individuals are updated at the same time based on these errors.

Optimizer

In previous research, the adversarial imitation learning has been modeled as the fitness function and the genetic algorithms is used; however, the gradient descent method is used here.

The reason for this is because it takes a lot of time to evolve many parameters using genetic algorithms. Another reason is that the genetic algorithms tend to change the structure drastically, which unstabilizes the co-evolution process. We update weights and biases of ANN in the direction of the minimization of the error.

Learning procedure

First, both two individuals generate their own song using their own initial value and the neural network. Then, each individual imitates the partner's song. They generate their songs while listening to the partner's song. Next, imitating-loss and imitated-loss about each individual is calculated. Finally, the artificial neural networks of both individuals are updated at the same time using the total of the two errors. By doing this in each epoch, learning process of each individual progresses.

Experiments

We performed the experiment of adversarial imitation learning in a variety of situations. First, we describe the results of the experiment in which two individuals imitate and are imitated by each other. Then, the results of the experiment in a variety of situations between two individuals are described. Finally the situations involving three and four individuals are described.

Experimental setting

The experiment is performed in the following experimental setup. The length of the generated time series T is 32, and the number of learning epochs is $1,000$. In order to avoid the exploding gradient problem when updating the initial value, the gradient clipping method is used.

The artificial neural networks of A and B are updated at the same time after the generation and imitation phases. The network consists of three layers, that is, input, hidden and output layers. The number of nodes N in the hidden layer is 32. The outputs of the network are limited to $[-1, 1]$ all the time because the node values of each layer are transformed by the tanh activation function.

Imitation game between two individual

The changes of loss and generated time series are shown in Fig. 4. The imitating-loss is smaller than the imitated-loss in both individuals. At the beginning of the learning, both individuals generate simple time series which fall into a fixed-point. After that, individual A sings a period-2 song and B's imitating song also becomes period-2. However, the original B song remains a fixed-point and the difference

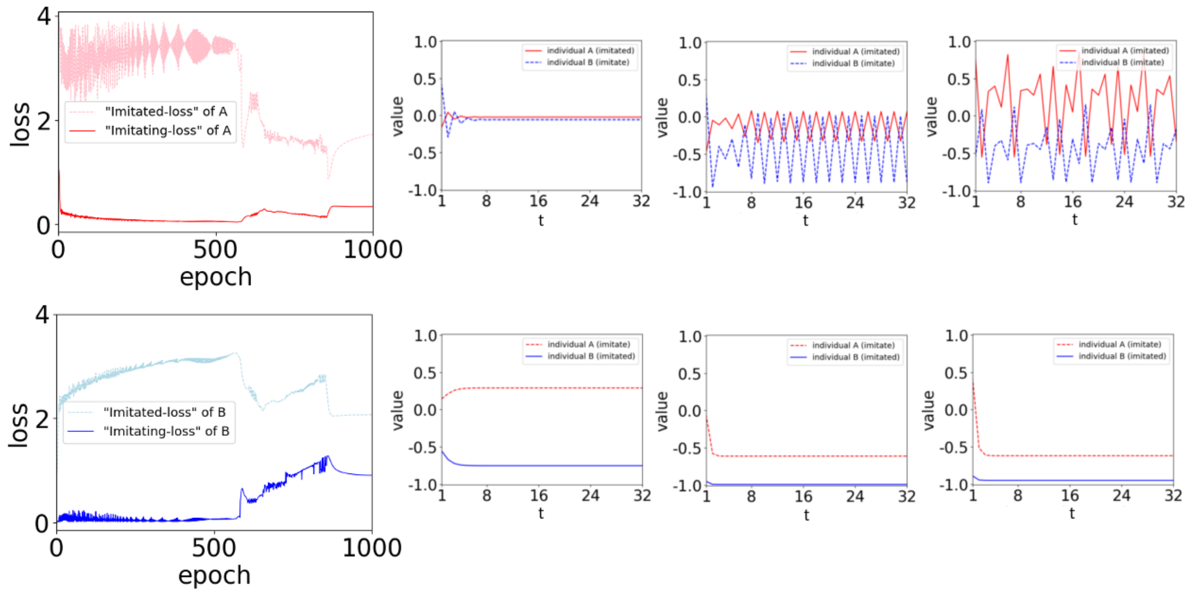

Figure 4: The leftmost column is each graph of changes in errors related to A and B. The remaining graphs are bird song time series of 1, 600 and 840 epoch from the left, respectively. The upper row is the case where A is imitated by B (B imitates A): the lower row is the case where B is imitated by A (A imitates B).

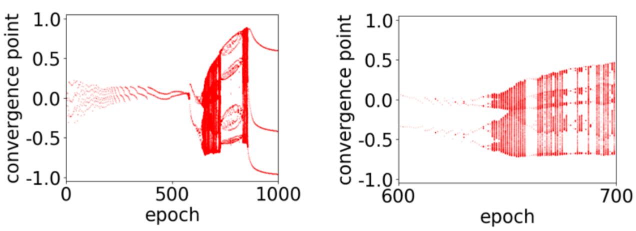

Figure 5: The bifurcation diagram of individual A. The left graph is the diagram from 0 to 1,000, and the right graph is the diagram from 600 to 700 which is an enlarged view of the left. The horizontal line shows the epoch.

between these songs when A imitates B becomes smaller than before. At an 840 epoch, A's original song shows more complex dynamics like chaos.

Figure. 5 is the bifurcation diagram. The vertical line shows the convergence point of the time series that is generated when individual A sings its own song by itself until it converges. The horizontal line shows an epoch, and from this figure, we can see at what point the dynamics has changed. The graph shows that A's song is very simple at the beginning, gradually increases in cycle. Further, because the song does not converge and the periodicity also disappears, it becomes like chaos in certain epochs through the learning process that occurs between the two singing individuals. The adversarial imitation learning appears to transform the songs to chaos.

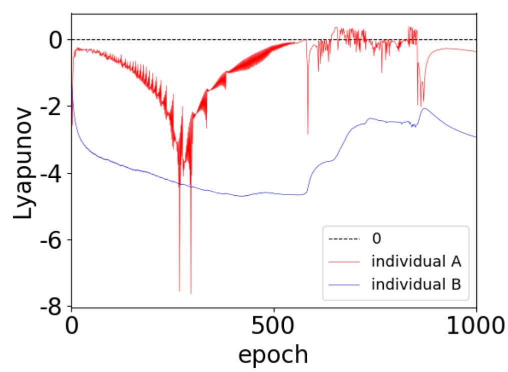

Figure 6: The transitions of the Lyapunov exponent of two individuals.

In order to evaluate the obtained bird songs quantitatively, the Lyapunov exponent λ is introduced (Lyapunov, 1992). λ is the degree to which a slight difference in the dynamic system is enlarged exponentially and is used to tell whether it is chaos. It is defined as follows:

$$\lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left| f'(x_i) \right|. \tag{14}$$

When $\lambda > 0$, it is a chaotic time series, but not a chaotic time series otherwise.

In this study, n is 1,000, and we calculate the Lyapunov exponent with the time series which is generated using the

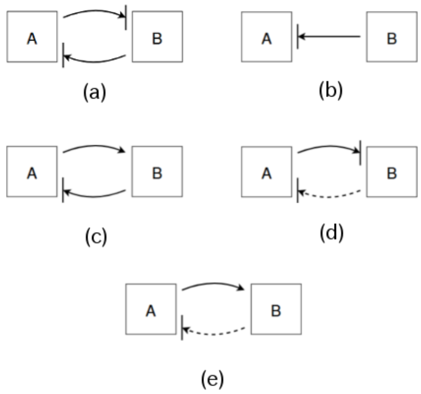

Figure 7: Different types of interactions for learning. See the main texts for details of the interaction.

Table 1: Frequency of generating a chaotic time series after 20 experiments. Each column of A, B indicates which individual generates the chaotic time series. The column of ratio is the probability that a chaotic time series will be generated by one individual.

interaction	A	B	ratio
(a)	5	3	0.200
(b)	0	0	0
(c)	0	0	0
(d)	4	0	0.100
(e)	0	0	0

initial value each individual has at the epoch. Furthermore, because the artificial neural network is expressed as eq (3), it can be differentiated.

The changes of the Lyapunov exponent are shown in Fig. 6. The very stable dynamics is destabilized gradually and becomes chaos. This chaos is sustained for a while. After that, the value stays around $\lambda = 0$. This might be the edge of chaos.

Results in different types of interaction

A chaotic time series is generated in the two individuals experiment, but it is not always observed that the birds' songs become chaotic behavior. In order to clarify what makes time series chaos, we perform experiments in different types of interactions between two individuals.

The types of interactions between two individuals are shown in Fig. 7. The arrow from A to B means that A imitates B's song and if the arrow is blocked with the line at the destination, it means that the destination individual performs not-being-imitated learning. The dotted line at the departure shows that the departure sings the songs by itself without imitation learning but the destination learns not to be imitated while listening to the song. Figure. 7 (a) shows that both individuals perform adversarial imitation learning as the experiment in the previous section. Figure. 7 (b) shows the unidirectional interaction where B imitates A's song and A learns not to be imitated and does not imitate B's song, that is, A learns using only imitated-loss, and B learns using only imitating-loss. In (c), A performs adversarial imitation learning, but only B imitates A's song without not-being-imitated learning. (d) is the opposite of (c). In (d), B performs not-being-imitated learning without imitation learning while A performs the adversarial imitation learning. In (e), either individual performs adversarial learning, that is, imitation and not-being-imitated learning, against the static network.

Table 2: Frequency of generating a chaotic time series in 20 experiments. Each column of A, B, C, and D means which individual generates a chaotic time series. The column of ratio is the probability that a chaotic time series is generated by one individual.

interaction	A	B	C	D	ratio
(a)	5	3	-	-	0.200
(f)	0	0	0	-	0
(g)	1	3	7	-	0.183
(h)	8	11	9	10	0.475

We repeated the experiment 20 times for each situation. The results are shown in Table 1. From these results, a chaotic time series is observed only it's in Fig. 7 (a) and (d). When both individuals perform adversarial imitation learning, the chaos is observed more often than that in other situations. Without adversarial imitation learning, the chaos is not observed. However, despite the fact that A performs the adversarial imitation learning, only (d) produced the chaos. The adversarial imitation learning might be necessary to produce the chaotic behaviors for an individual but not-being-imitated learning might be also necessary to unstabilize the dynamics as an interaction.

Results under more social situation

The chaotic dynamics through the adversarial imitation learning were observed, but rarely. What we want to clarify in this section is what encourages the emergence of chaos. Usually, social interaction is useful for such an outcome. Here, we increased the number of individuals in the simulation.

The type of interactions between three and four are shown in Fig. 8. The arrow from A to B and the block are the same as those in Fig. 7. All artificial networks are updated at the same time in the same manner as the previous experiments.

Figure. 8 (f) shows the situation where all three individuals perform adversarial imitation learning but the relation is not direct but indirect: (g) and (h) are the relations with the complete graph with three and four individuals, respectively

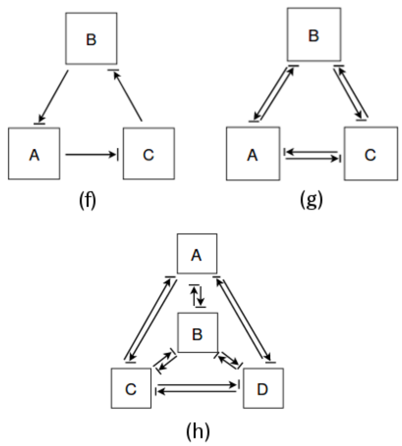

Figure 8: The situations of three and four individuals. The arrow is same in the case of Fig. 7.

to see if the number of individuals increase the probability of the emergence of chaos.

We performed the experiment 20 times corresponding to each situation. The results of these experiments are shown in Table 2. The results indicate that generating a chaotic time series needs direct adversarial imitation learning. Moreover, the interactions among four individuals cause chaotic behaviors more often.

Discussion

This paper suggests that adversarial imitation learning, which is modeled as the learning where an individual must imitate the other and must not be imitated by the other at the same time, prompts a time series to generate to be complex. These time series change from an orbit that converges to a fixed point to a periodic one, and further to a chaotic one as learning progresses. However, there was a case where a chaotic time series is not generated in a model with two individuals.

Moran and Pollack present the coevolutionary dynamics of complexity growth in a variety of multi-species simulations by a model of game players based on finite state automata (Moran and Pollack, 2017). Hence, there is the possibility that particular situations prompt an increase in complexity. Thus, by changing situations when individuals build adversarial imitation relationship, we aimed to increase the frequency of generating a chaotic time series. The results showed that the more the individual builds an adversarial imitation relationship with other individuals, the more easily the corresponding time series becomes chaos. Considering a situation where real birds are exposed to an environment in which there are many bird songs, this conclusion is reasonable.

In this study, an ANN is updated by two contradictory loss functions, which are imitating-loss and imitated-loss, which is actually similar to a GAN (Generative Adversarial Network.) (Salimans et al., 2016). As with a GAN, the competing loss function might help the network to develop complex structures.

Conclusion

This study has shown that adversarial imitation relationships lead to chaos time series and that a specific interaction advances it. However, in this paper, a model of adversarial imitation learning involving a few individuals is presented. Therefore, further consideration will be needed to yield any findings about an experimental setting that is closer to the vocal activities of real birds and that which involves more individuals. Also, Okanoya showed that a Bengalese finch's songs have an underlying grammar. Therefore, considering the structure of a recurrent neural network (Hopfield, 1982), we not only expect emergence of chaos but also emergence of a grammar related to a bird song.

References

Catchpole, C. K. (1981). Vocal communication in birds. *Japanese Journal of Ornithology*, 30(2-3):87–89.

Chauvin, Y. and Rumelhart, D. E. (1995). *Backpropagation: theory, architectures, and applications*. Psychology Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Lyapunov, A. M. (1992). The general problem of the stability of motion. *International journal of control*, 55(3):531–534.

Moran, N. and Pollack, J. (2017). Effects of cooperative and competitive coevolution on complexity in a linguistic prediction game. In *Proceedings of the 14th European conference on artificial life*, pages 298–205.

Robinson, A. and Fallside, F. (1987). *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242.

Suzuki, J. and Kaneko, K. (1994). Imitation games. *Physica D: Nonlinear Phenomena*, 75(1-3):328–342.

ten Cate, C. and Okanoya, K. (2012). Revisiting the syntactic abilities of non-human animals: natural vocalizations and artificial grammar learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1598):1984–1994.