scientific reports

OPEN

Arm order recognition in multi-armed bandit problem with laser chaos time series

Naoki Narisawa[1]✉, Nicolas Chauvet[1,2], Mikio Hasegawa[3] & Makoto Naruse[1,2]✉

By exploiting ultrafast and irregular time series generated by lasers with delayed feedback, we have previously demonstrated a scalable algorithm to solve multi-armed bandit (MAB) problems utilizing the time-division multiplexing of laser chaos time series. Although the algorithm detects the arm with the highest reward expectation, the correct recognition of the order of arms in terms of reward expectations is not achievable. Here, we present an algorithm where the degree of exploration is adaptively controlled based on confidence intervals that represent the estimation accuracy of reward expectations. We have demonstrated numerically that our approach did improve arm order recognition accuracy significantly, along with reduced dependence on reward environments, and the total reward is almost maintained compared with conventional MAB methods. This study applies to sectors where the order information is critical, such as efficient allocation of resources in information and communications technology.

Chaos can be defined as random oscillations generated by deterministic dynamics[1]. Chaotic time series are very sensitive to initial conditions, which render long-term predictions unachievable unless infinite observation accuracy is attained in the beginning[2]. The close relationship between lasers and chaos has been known for a long time; the output of a laser generates chaotic oscillations when a time-delayed optical feedback is injected back into the laser cavity[3]. Laser chaos exhibits ultrafast dynamics beyond GHz regime/domain; hence, various engineering applications have been examined in the literature. Examples range from optical secure communication[2] and fast physical random bit generation[4] to secure key distribution using correlated randomness[5].

The present study relates to the application of laser chaos to a multi-armed bandit problem (MAB)[6]. Reinforcement learning (RL), a branch of machine learning along with supervised and unsupervised learning, studies optimal decision-making rules. It differs from other machine learning tasks (e.g. image recognition) as the notion of reward comes into play in RL. The goal of RL is to construct decision-making rules that maximize obtained rewards; hence, gaming AI is a well-known application of RL[7]. In 2015, AlphaGo, developed by Google DeepMind, defeated a human professional Go player for the first time[8].

The MAB is a sequential decision problem of maximizing total rewards where there are $K(>1)$ arms, or selections, whose reward probability is unknown. The MAB is one of the simplest problems in RL. In a MAB, a player can receive reward information that pertains only to the selected arm at each time step, so a player cannot obtain the reward information for a non-selected arm. The MAB exhibits a trade-off between exploration and exploitation. Sufficient exploration is necessary to estimate the best arm more accurately, but it accompanies low-reward arm selections. Hence, excessive exploration can lead to significant losses. Furthermore, to maximize rewards, one needs to choose the best arm (use the exploitation principle). However, if the search for the best arm fails, then a non-best option may be mistakenly chosen very likely. Therefore, it is important to balance exploration and exploitation.

An algorithm for the MAB using laser chaos time series has been proposed in 2018[6]. This algorithm sets two goals: to maximize the total rewards and to identify the best arm. However, concerning real-world applications, maximizing the rewards and finding the optimal arm may not be enough to solve a problem. For example, there is a study to improve communication throughput by treating the channel selection in wireless communications as a MAB[9]. Should we have multiple channel users, not all users can use the best channel simultaneously; accordingly,

[1]Department of Mathematical Engineering and Information Physics, Faculty of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan. [2]Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan. [3]Department of Electrical Engineering, Tokyo University of Science, 6-3-1 Niijuku, Katsushika-ku, Tokyo 125-8585, Japan. ✉email: narisawa-naoki682@g.ecc.u-tokyo.ac.jp; makoto_naruse@ipc.i.u-tokyo.ac.jp

https://doi.org/10.1038/s41598-021-83726-8

there may be situations where compromises must be made, i.e., other channels will be selected. Now it is obvious that particular channel performance ranking information would be useful when considering non-best channels.

Conversely, when there are no other users, a player (the single user) can simultaneously utilize top-ranking options to accelerate the communication ability, similar to the channel bonding in local area networks[10]. The purpose of this study is to accurately recognize the order of the expected rewards of different arms using a chaotic laser time series and to minimize the reduction of accumulated rewards due to too detailed exploration.

Principles

Definition and assumption.

We consider a MAB problem in which a player selects one of K slot machines, where $K = 2^M$ and M is a natural number. The K slot machines are distinguished by identities numbered from 0 to $K-1$, which are also represented in M-bit binary code given by $S_1 S_2 \ldots S_M$ with $S_i \in \{0,1\}$ $(i = 1, \ldots, M)$. For example, when $K = 8$ (or $M = 3$), the slot machines are numbered by $S_1 S_2 S_3 = \{000, 001, \ldots, 110, 111\}$. In this study, we assume that $\mu_i \neq \mu_j$ if $i \neq j$, and we define the k-th max and k-th argmax operators as $\max^k\{\}$ and $\arg\max^k\{\}$. The variables used in the study are defined as described below:

- $X_i(n)$: Obtained reward from arm i at time step n independent at each time step. $x_i(n)$ is observed value.
- $\mu_i := \mathbb{E}[X_i(n)]$. (Consistent regardless of time step)
- $\mu^* := \max_i \mu_i, \quad i^* := \arg\max_i \mu_i$
- $T_i(n)$: Number of selections of arm i by the end of time step n. $t_i(n)$ is observed value.
- $A(n)$: Arm selected at time step n. $a(n)$ is the observed value.
- $[k] := \arg\max_i^k \mu_i$: k-th best arm.

We estimate the arm order of reward expectations by calculating the sample mean of the accumulated reward at each time step. Specifically, the sample means of rewards obtained from arm i by time step n is calculated as follows:

$$\hat{\mu}_i(n) = \frac{R_i(n)}{T_i(n)}, \quad \text{where} \quad R_i(n) := \sum_{s=1}^{n} X_i(s) \cdot \mathbb{I}[A(s) = i] \tag{1}$$

In each time step n, we estimated the arm $j := \arg\max_i^k \hat{\mu}_i(n)$ as the k-th best arm.

Time-division multiplexing of laser chaos.

The proposed method is based on the MAB algorithm reported in 2018[6]. This method consists of the following steps: [STEP 1] decision making for each bit of the slot machines, [STEP 2] playing the selected slot machine, and [STEP 3] updating the threshold values.

[STEP 1] Decision for each bit of the slot machine. First, the chaotic signal $s(t_1)$ measured at $t = t_1$ is compared to a threshold value denoted as TH_1. If $s(t_1) \geq TH_1$, then bit S_1 is assigned 1. Otherwise, S_1 is assigned 0. To determine the value of S_k $(k = 2, \ldots, M)$, the chaotic signal $s(t_k)$ measured at $t = t_k (> t_{k-1})$ is compared to a threshold value denoted as $TH_{k,S_1 \ldots S_{k-1}}$. If $s(t_k) \geq TH_{k,S_1 \ldots S_{k-1}}$, then bit S_k is assigned 1. Otherwise, S_k is assigned 0. After this process, a slot machine with the number represented in a binary code $S_1 \ldots S_M$ is selected.

[STEP 2] Slot machine play. Play the selected slot machine.

[STEP 3] Threshold values adjustment. If the selected slot machine yields a reward, then the threshold values are adjusted in a way that the same decision will be more likely to be selected. For example, if S_1 is assigned 0 and the player gets a reward, then TH_1 should be increased because doing so increases the likelihood of getting $S_1 = 0$ again. All of the other threshold values involved in determining the decision (i.e. $TH_{2,S_1}, \ldots, TH_{M,S_1 \ldots S_{M-1}}$) are updated in the same manner.

If the selected slot machine does not yield a reward, then the threshold values are adjusted to make the same decision less likely to take place. For example, if S_1 is assigned 1 and the player does not get a reward, then TH_1 should be increased because of the decreased likelihood of getting $S_1 = 1$. Again, all of the other threshold values involved in determining the decision (i.e. $TH_{2,S_1}, \ldots, TH_{M,S_1 \ldots S_{M-1}}$) are updated in the same manner.

Arm order recognition algorithm with confidence intervals.

Confidence intervals. An overview of our proposed algorithm is shown in Fig. 1a. For each threshold value $TH_{j,b_1 \ldots b_{j-1}}$ $(j \in \{1, \ldots, M\}$, $b_1, \ldots, b_{j-1} \in \{0,1\})$ and $z \in \{0,1\}$, the following values $\hat{P}(z;n)$ and $C(z;n)$ are calculated:

$$\hat{P}_{j,b_1 \ldots b_{j-1}}(z;n) := \frac{\sum_{i \in I_{j,b_1 \ldots b_{j-1}}(z)} R_i(n)}{\sum_{i \in I_{j,b_1 \ldots b_{j-1}}(z)} T_i(n)}, \quad C_{j,b_1 \ldots b_{j-1}}(z;n) := \gamma \cdot \sqrt{\frac{\log n}{\sum_{i \in I_{j,b_1 \ldots b_{j-1}}(z)} T_i(n)}} \tag{2}$$

$$I_{j,b_1 \ldots b_{j-1}}(z) := \begin{cases} \{i \mid \text{machine } i \text{ is available if } s(t_j) \geq TH_{j,b_1 \ldots b_{j-1}}\} & (\text{if } z = 1) \\ \{i \mid \text{machine } i \text{ is available if } s(t_j) < TH_{j,b_1 \ldots b_{j-1}}\} & (\text{if } z = 0) \end{cases} \tag{3}$$

$I_{j,b_1 \ldots b_{j-1}}(z)$ represents a subset of machine arms. If machine i can be selected when the signal $s(t_j)$ is more than $TH_{j,b_1 \ldots b_{j-1}}$, then i is included in $I_{j,b_1 \ldots b_{j-1}}(1)$. Otherwise, i is not included in $I_{j,b_1 \ldots b_{j-1}}(1)$. In the same way, if

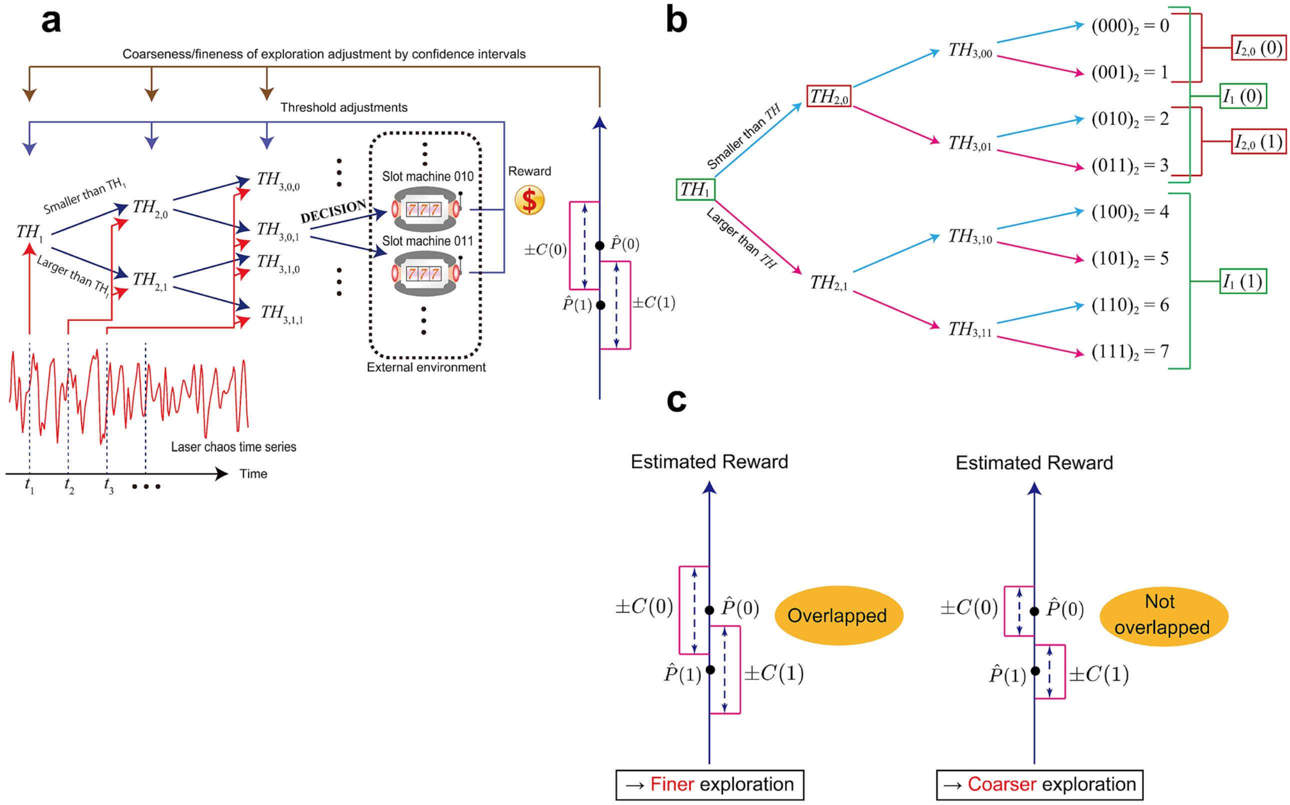

Figure 1. Architecture of the proposed method with confidence intervals. (**a**) Solving the MAB with $K = 2^M$ arms using a pipelined arrangement of comparisons between thresholds and a series of chaotic signal sequences. (**b**) Correspondence between threshold value TH and a subset of arms $I(z)$ $(z \in \{0, 1\})$ in the example of an eight-armed bandit problem. For each threshold TH_*, two types of arm set $I_*(0)$ and $I_*(1)$ are defined. (**c**) Coarseness/fineness of exploration adjustment by confidence intervals. For each threshold TH_*, the fineness of the threshold adjustment is changed depending on whether two confidence intervals $\hat{P}_*(0; n) \pm C_*(0; n)$ and $\hat{P}_*(1; n) \pm C_*(1; n)$ are overlapped. A part of images in (**a**) is adapted from Naruse et al., Sci. Rep. 8, 10890 (2018). Copyright 2018 Author(s), licensed under a Creative Commons Attribution 4.0 License.

machine i can be selected when the signal $s(t_j)$ is less than or equal to $TH_{j,b_1\ldots b_{j-1}}$, then i is included in $I_{j,b_1\ldots b_{j-1}}(0)$. Otherwise, i is not included in $I_{j,b_1\ldots b_{j-1}}(0)$. For example, in the case of an eight-armed bandit problem (Fig. 1b):

$$I_1(0) = \{0, 1, 2, 3\}, \quad I_{2,0}(0) = \{0, 1\}, \quad I_{2,1}(1) = \{3, 4\}, \quad I_{3,00}(1) = \{1\}.$$

$\hat{P}_{j,b_1\ldots b_{j-1}}(z; n)$ represents the sample means of rewards obtained from machines in $I_{j,b_1\ldots b_{j-1}}(z)$. $C_{j,b_1\ldots b_{j-1}}(z; n)$ represents the confidence interval width of the estimated value $\hat{P}_{j,b_1\ldots b_{j-1}}(z; n)$. The lower $C(z; n)$, the higher the estimation accuracy. Parameter γ indicates the degree of exploration : a higher γ means that more exploration is needed to reach a given confidence interval width.

Coarseness/fineness of exploration adjustments by confidence intervals. At each threshold $TH_{j,b_1\ldots b_{j-1}}$, if the two intervals

$$\left[\hat{P}(0; n) - C(0; n),\ \hat{P}(0; n) + C(0; n)\right] \quad \text{and} \quad \left[\hat{P}(1; n) - C(1; n),\ \hat{P}(1; n) + C(1; n)\right]$$

are overlapped, we suppose there is a likelihood of a change in the order relationship between $\hat{P}(0; n)$ and $\hat{P}(1; n)$; that is, the order of $\hat{P}(0; n)$ and $\hat{P}(1; n)$ is not known yet. Therefore, the exploration process should be executed more carefully. Hence, the threshold value should be closer to 0, which is a balanced situation, or we should perform further exploration, so that the threshold adjustment becomes finer. Conversely, if the two intervals are not overlapped, then we suppose a low likelihood of a wrong estimate of the order relationship between $\hat{P}(0; n)$ and $\hat{P}(1; n)$. Hence, we should continue exploration more coarsely so that the threshold adjustment will be accelerated (Fig. 1c).

Results

Experimental settings.

We have evaluated the performance of the methods for two cases: a four-armed bandit and an eight-armed bandit. First, the reward probability of each arm is assumed to follow the Bernoulli distribution: $\Pr(X_i = x) = \mu_i^x (1 - \mu_i)^{1-x}$. Each reward environment $\nu := (\mu_0, \ldots, \mu_{K-1})$ is set to satisfy the

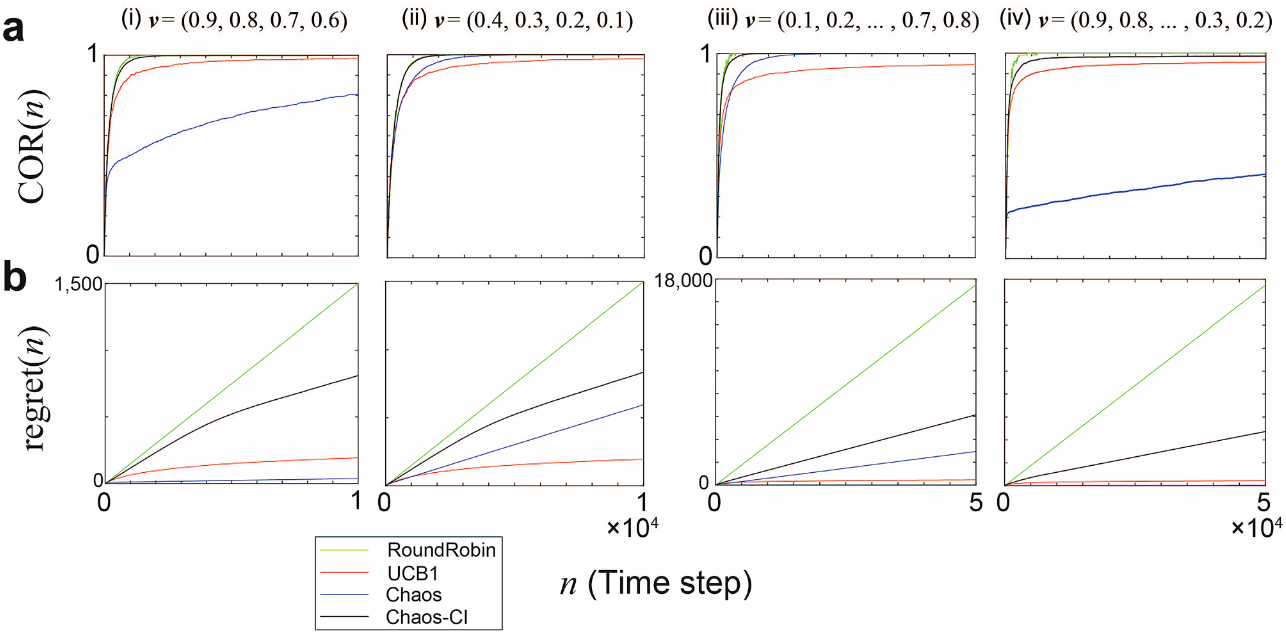

Figure 2. Time evolution of metrics COR(n) and regret(n) under one reward environment ν. Each column shows the results under each environment: the left two columns are in the four-armed bandit and the right two columns are in the eight-armed bandit. (**a**) COR(n) calculated as Eq. (6). (**b**) regret(n) calculated as Eq. (5).

following conditions: (i) $\forall i : \mu_i \in \{0.1, 0.2, \ldots, 0.8, 0.9\}$, (ii) $i \neq j \Rightarrow \mu_i \neq \mu_j$. In this experiment, a variety of assignments of reward probabilities ν satisfying the above conditions were prepared, and the performance was evaluated under every reward environment ν. We have defined the reward, regret, and correct order rate (COR) as metrics to quantitatively evaluate the performance of the method.

$$\mathrm{reward}(n) := \frac{1}{l_m}\sum_{l=1}^{l_m}\left[\sum_{s=1}^{n} x^{(l)}_{a^{(l)}(s)}(s)\right], \tag{4}$$

$$\mathrm{regret}(n) := \frac{1}{l_m}\sum_{l=1}^{l_m}\left[\sum_{i\neq i^*}(\mu^* - \mu_i)t_i^{(l)}(n)\right], \tag{5}$$

$$\mathrm{COR}(n) := \frac{1}{l_m}\sum_{l=1}^{l_m}\mathbb{I}\left[\bigcap_{k=1,\ldots,4}\left\{\arg\overset{k}{\max_{i=1,\ldots,K}}\hat{\mu}_i^{(l)}(n) = [k]\right\}\right] \tag{6}$$

where n denotes number of time steps, $t_i(n)$ is the number of selections of arm i up to time step n, and l_m represents the number of measurements in one reward environment ν. For the accuracy of arm order recognition, we considered the estimation accuracy of the top four arms regardless of the total number of arms. We prepared all 144 reward environments ν (all combinations satisfying the above conditions and $\max_{i\neq j}|\mu_i - \mu_j| = 0.3$) for the four-armed bandit problems and 100 randomly selected reward environments for the eight-armed bandit problems. The performances of four methods were compared: **RoundRobin** (all arms are selected in order at each time step), **UCB1** (method for maximizing the total rewards proposed in 2002[11]), **Chaos** (previous method using the laser chaos time series[6], only finding the best arm, not recognizing the order), and **Chaos-CI** (proposed method using laser chaos time series and with confidence intervals). The details of UCB1 used in the present study are described in the *Methods* section. The purpose of this study is to extend the existing Chaos method to recognize the arm order. We should consider the trade-off between order recognition and reward maximization. As introduced above, RoundRobin and UCB1 were considered to examine quantitative performance analysis. RoundRobin systematically accomplishes the order recognition whereas UCB1 is known to achieve $O(\log n)$ regret at time n. We consider that these are appropriate and contrasting representative methods in the literature to examine the trade-off and the essential interest of the present study. Meanwhile, comparison to other bandit algorithms such as Thompson sampling[12] or arm elimination[13] is expected to trigger stimulating future discussions, leading to further improvement of the proposed Chaos-CI algorithm.

Evaluation under one reward environment. The curves in Fig. 2a and b show the time evolutions of regret(n) and COR(n), respectively, over $l_m = 12,000$ measurements under specific reward environments $\nu = (\mu_0, \ldots, \mu_{K-1})$. Specifically, columns (i) and (ii) pertain to the four-armed bandit problems defined by $\nu = (0.9, 0.8, 0.7, 0.6)$

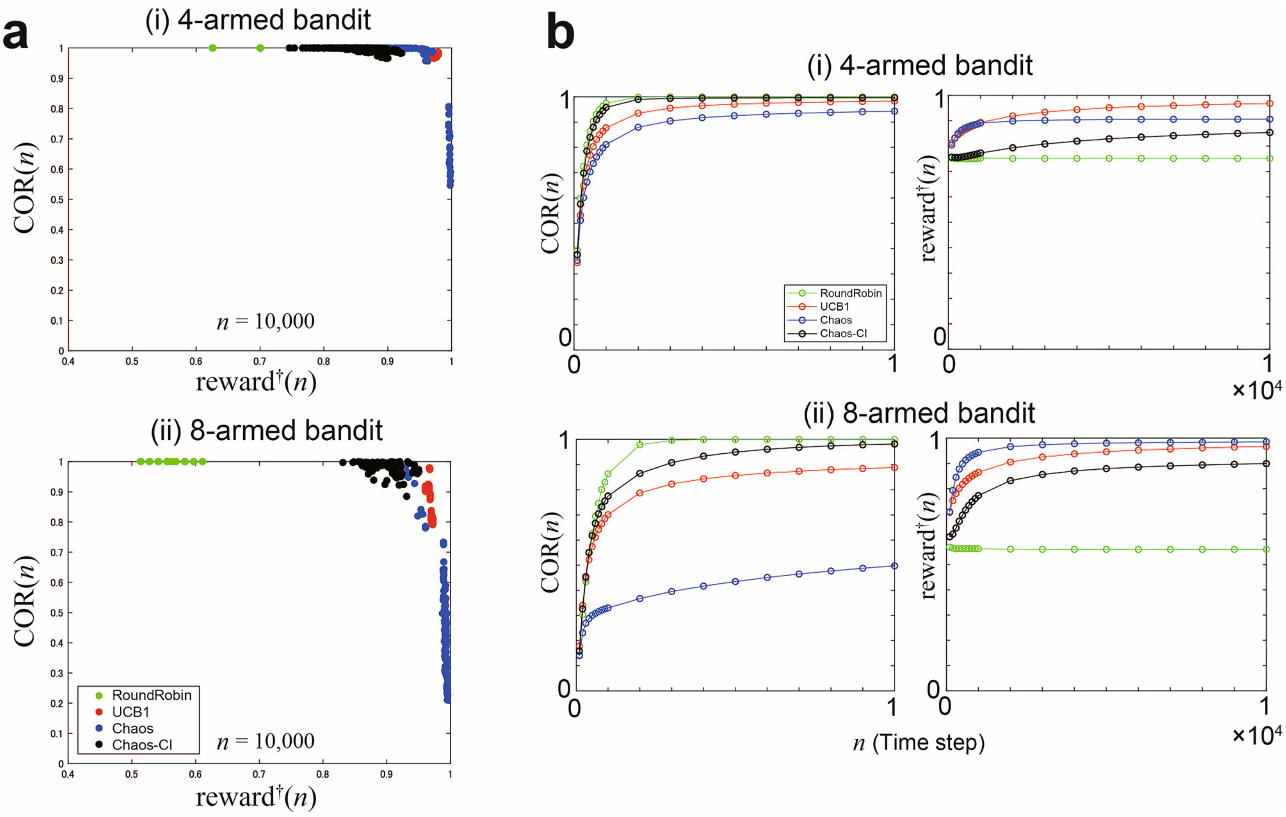

Figure 3. Metrics over the whole reward environments prepared. (**a**) Each scatter plot represents a normalized reward $\text{reward}^\dagger(n)$ and correct order rate $\text{COR}(n)$ at time step $n = 10{,}000$ under one reward environment ν: $(\text{reward}^\dagger_\nu(10{,}000), \text{COR}_\nu(10{,}000))$. The more the scatter plot is at the top of the graph, the higher the order estimation accuracy is, and the more the scatter plot is at the right, the greater the obtained reward is. (**b**) Time evolution of the average value of each metric over the whole ensemble of reward environments ($1 \leq n \leq 10{,}000$).

and $\nu = (0.4, 0.3, 0.2, 0.1)$, whereas columns (iii) and (iv) depict the eight-armed bandit problems given by $\nu = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)$ and $\nu = (0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2)$. The curves were colour coded for an easy method comparison. In the arm order recognition, Chaos-CI and RoundRobin presented high accuracy in the early time step. In terms of total reward, Chaos and UCB1 achieved the greatest rewards. The post-convergence behavior of Chaos and Chaos-CI is not necessarily the same. The reason is that the parameters Λ and Ω that determine the scale of threshold change are fixed values in Chaos, whereas they change adaptively according to past reward information in Chaos-CI.

Evaluation of the whole reward environments. Figure 3a summarizes the relationship between total rewards and order estimation accuracy: x-axis represents the normalized reward $\text{reward}^\dagger(n)$, whereas y-axis represents the COR $\text{COR}(n)$. Here, a normalized reward is defined as follows:

$$\text{reward}^\dagger(n) := \frac{\text{reward}(n)}{\mu^* \cdot n}.$$

Each plot in the graph indicates $\text{reward}^\dagger(n)$ and $\text{COR}(n)$ at time step $n = 10{,}000$ under one reward environment ν:

$$(\text{reward}^\dagger_\nu(10{,}000), \text{COR}_\nu(10{,}000)).$$

Figure 3b shows the time evolution of the average value of each metric over the whole ensemble of reward environments from $n = 1$ to $n = 10{,}000$:

$$\frac{\sum_\nu \text{COR}_\nu(n)}{\sum_\nu 1}, \quad \frac{\sum_\nu \text{reward}^\dagger_\nu(n)}{\sum_\nu 1}, \quad (1 \leq n \leq 10{,}000).$$

Compared to UCB1, Chaos-CI can recognize the arm order faster. On the other hand, Chaos-CI can get more rewards than RoundRobin.

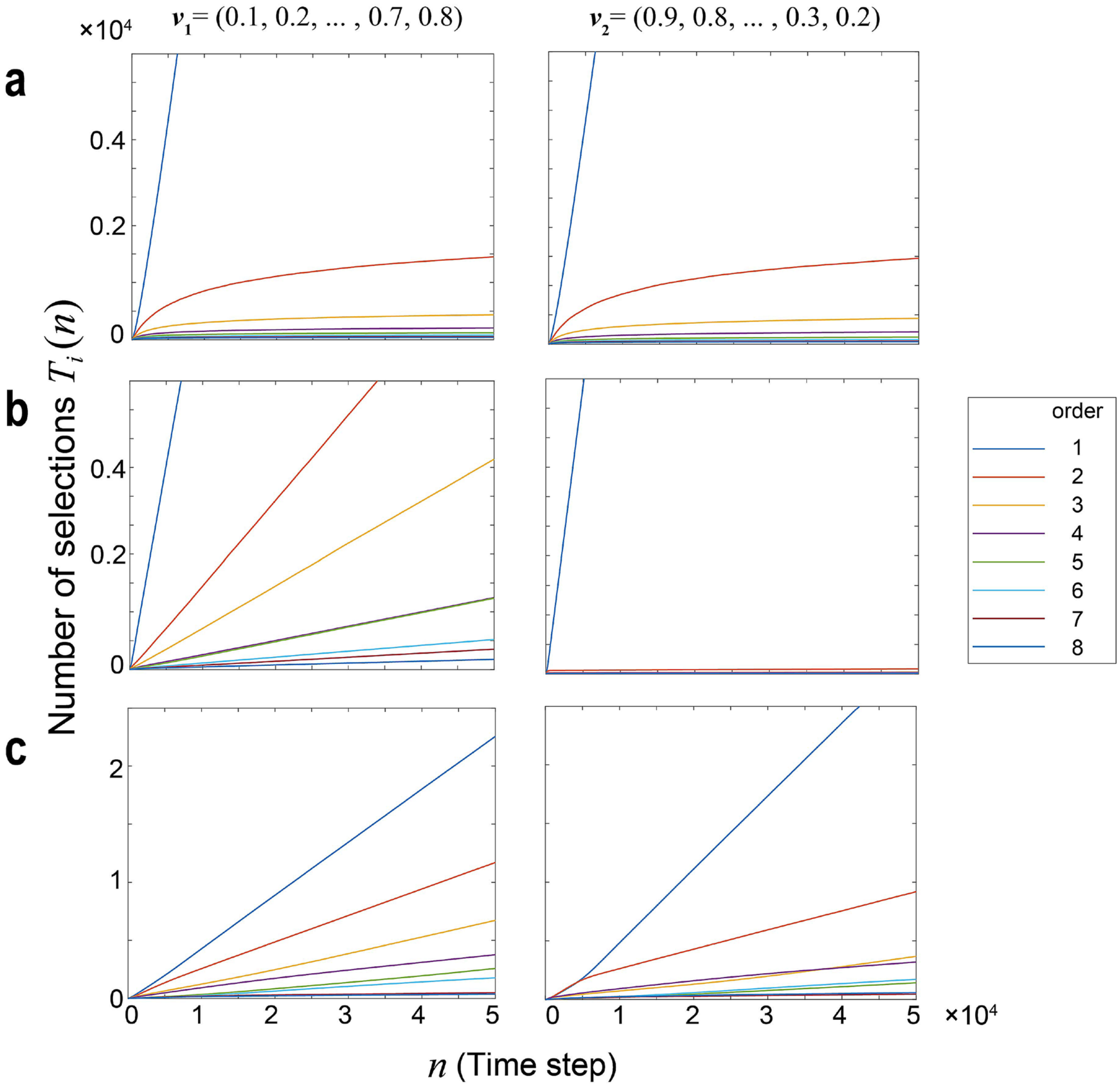

Figure 4. Time evolution of the selection number of each arm for three methods: UCB1, Chaos, and Chaos-CI. The three figures on the left represent $T_i(n)$ under reward environment ν_1, and the three on the right represent under ν_2. Each row represents the result of each method: (**a**) UCB1, (**b**) Chaos, and (**c**) Chaos-CI.

Discussion

Difficulty of maximizing rewards and arm order recognition. The results of the numerical simulations on the four-armed and eight-armed bandit problems show similar trends: there is a trade-off between the maximized total rewards and arm order recognition. As RoundRobin selects all arms equally, we always achieve a perfect COR at a time step $n = 10{,}000$ for any given reward environment. However, we cannot maximize rewards because regret linearly increases with time. On the contrary, in Chaos, we achieved normalized rewards of almost unity at the time step of $n = 10{,}000$ with respect to many types of reward environments. However, we can observe inferior performances regarding the arm order recognition accuracy because the arm selection is greatly biased to the best arm. In terms of the COR, the COR on RoundRobin and Chaos-CI (proposed method) quickly converged to unity. In terms of the total rewards, Chaos (previous method) and UCB1 are more active in using the exploitation principle to obtain greater rewards. The proposed method, Chaos-CI, achieves an outstanding performance on the arm order recognition and reward.

Number of arm selections. Figure 4a, b, and c show the time evolutions of $T_i(n)$ by UCB1, Chaos and Chaos-CI, respectively (RoundRobin leading to equal number of selections for all arms at any time). Here, we examine the two types of reward environments ν_1 and ν_2 in an eight-armed bandit given by $\nu_1 := (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)$ and $\nu_2 := (0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2)$ corresponding to the left and right columns of Fig. 4.

This figure shows that the selection number of the best arm (i.e. $T_{[1]}(n)$) increases by $O(n)$ and $T_i(n) (i \neq i^*)$ increases almost by $O(\log n)$ in UCB1. Through the evolution of $T_i(n)$, UCB1 can achieve a regret of $O(\log n)$, but the convergence of $\mathrm{COR}(n)$ is slow. In the proposed Chaos-CI, the selection number of every arm evolves

	COR		reward†	
	Average	Variance	Average	Variance ($\times 10^{-3}$)
RoundRobin	1	0	0.560	0.8852
UCB1	0.888	0.0026	0.967	0.0116
Chaos	0.497	0.0413	0.985	0.3073
Chaos-CI	0.982	0.0004	0.899	0.6140

Table 1. Sample average and variance of metrics over 100 reward environments.($n = 10{,}000$)

in a linear order. Therefore, the arm order recognition accuracy is faster than UCB1. Although the selections of non-top arms in the linear order cause regret to increase in a linear order, the slope of the linear-order regret is significantly decreased compared with that of RoundRobin by selecting better arms more often or by prioritizing the search (i.e. $T_{[1]}(n) > \cdots > T_{[K]}(n)$).

Environment dependency. As shown in Figs. 3 and 4, the performances of Chaos are very different depending on reward environments ν_1 and ν_2. This finding is clearly linked with the arm selection number $T_i(n)$. In reward environment ν_1, all $T_i(n)$ evolve in a linear order, but in reward environment ν_2, $T_i(n)(i \neq i^*)$ is approximately 100 at time step $n = 50{,}000$. Thus, the performance of Chaos heavily depends on the given reward environment. Table 1 summarizes the sample variance of metrics over 100 reward environments in an eight-armed bandit. Our proposed algorithm (Chaos-CI) depends on the order of the arms. If the difference between the average reward of each branch is large, we consider that the ranking estimation is easy. However, from Fig. 3(a) and Table 1, we observe that the proposed method always estimates the rankings with high accuracy regardless of the order of the arms. This estimation accuracy outperforms the performance of UCB1, which is an algorithm that does not depend on the order of the arms. In terms of obtained rewards, Chaos-CI has a larger variance than UCB1 and Chaos but is more stable than RoundRobin.

In the experiments, the expected reward μ_i is limited so that the difference in the estimated difficulty does not vary drastically from problem to problem because the larger the difference in the expected reward value of each arm, the easier the problem becomes to solve. Meanwhile, if the difference in reward expectation for each arm becomes even smaller (specifically, smaller than 0.1), the correct order recognition in its exact sense will be significantly challenging. At the same time though, such a case means that there are not significant reward differences regardless of which arm is pulled. Hence, we consider that the evaluation method or the definition of correct order recognition may need revision. We expect these points to form the basis of interesting future studies.

On the other hand, if the reward distribution is more diverse, for example, [0.95, 0.9, 0.6, 0.5], UCB1, which aims to maximize the cumulative reward, will stop selecting the lower-reward-probability arms at an early exploration phase, leading to the degradation of the accuracy of rank recognition. Conversely, since the proposed method adjusts the thresholds based on the confidence intervals in all branches, it is expected that the rank recognition accuracy will not be degraded.

Conclusions

In this study, we have examined ultrafast decision making with laser chaos time series in reinforcement learning (e.g. MAB) and set a goal to recognize the arm order of reward expectations by expanding the previous method, that is, time-division multiplexing of laser chaos recordings. In the proposed method, we have introduced exploration-degree adjustments based on confidence intervals of estimated rewards. The results of the numerical simulations based on experimental time series show that the selection number of each arm increases linearly, leading to a high and rapid order recognition accuracy. Furthermore, arms with higher reward expectations are selected more frequently; hence, the slope of regret is reduced, although the selection number of an arm still linearly increases. Compared with UCB1 and Chaos, Chaos-CI (proposed method) is less dependent on the reward environment, indicating its potential significance in terms of robustness to environmental changes. In other words, Chaos-CI can make more accurate and stable estimates of arm order. Meanwhile, expressing the accuracy of rank estimation in terms of earned rewards in a single metric is an interesting, important, and challenging problem. We plan to explore this in our future research. Such an order recognition is useful in applications, such as channel selection and resource allocation in information and communications technology, where compromise actions or intelligent arbitrations are expected.

Methods

Optical system.

The device used was a distributed feedback semiconductor laser mounted on a butterfly package with optical fibre pigtails (NTT Electronics, KELD1C5GAAA). The injection current of the semiconductor laser was set to 58.5 mA ($5.37I_{th}$), where the lasing threshold I_{th} was 10.9 mA. The relaxation oscillation frequency of the laser was 6.5 GHz, and its temperature was maintained at 294.83 K. The optical output power was 13.2 mW. The laser was connected to a variable fibred reflector through a fibre coupler, where a fraction of light was reflected back to the laser, generating high-frequency chaotic oscillations of optical intensity[3,14,15]. The length of the fibre between the laser and reflector was 4.55 m, corresponding to a feedback delay time (round trip) of 43.8 ns. Polarization maintaining fibres were used for all of the optical fibre components. The optical signal was detected by a photodetector (New Focus, 1474-A, 38 GHz bandwidth) and sampled using a digital oscilloscope (Tektronics, DPO73304D, 33 GHz bandwidth, 100 GSample/s, eight-bit vertical resolution). The RF spectrum of the laser was measured by an RF spectrum analyzer (Agilent, N9010A-544, 44 GHz bandwidth).

USB1 algorithm. In UCB1[11], we select the arm j that maximize the score based on

$$\bar{X}_j(n) + \sqrt{\frac{2\log n}{T_j(n)}} \tag{7}$$

where $\bar{X}_j(n)$ is the expected average reward obtained from arm j, $T_j(n)$ is the number of times machine j is played so far, and n is the total numbers of plays so far.

Details of the time-division multiplexing algorithm.

Algorithm 1 Time-division multiplexing of laser-chaos[6]

```
Parameters: $0 < \alpha < 1, \quad \Delta_S > 0, \quad \Delta_L > 0, \quad \Lambda > 0, \quad \Omega > 0$
Initialization: $\tau_S \leftarrow \tau_{init}, \quad \forall m, S_1, \ldots, S_{m-1} : TH_{m,S_1 \cdots S_{m-1}} \leftarrow 0$
for $n = 1, \ldots, n_{\max}$ do
    $\tau_L \leftarrow \tau_S$
    % STEP1
    $S_1 \leftarrow \mathbb{I}[s(\tau_L) \geq TH_1]$
    for $m = 2, \ldots, M$ do
        $\tau_L \leftarrow \tau_L + \Delta_L$
        $S_m \leftarrow \mathbb{I}\left[s(\tau_L) \geq TH_{m,S_1 \cdots S_{m-1}}\right]$
    end for
    % STEP2
    $a \leftarrow (S_1 \cdots S_M)_2$
    sample $x_a(n) \sim P_a$
    % STEP3
    if $x_a(n) > 0$ then
        for $m = 1, \ldots, M$ do
            $TH_{m,S_1 \cdots S_{m-1}} \leftarrow \alpha TH_{m,S_1 \cdots S_{m-1}} + \Lambda \cdot (-1)^{S_m}$
        end for
    else
        for $m = 1, \ldots, M$ do
            $TH_{m,S_1 \cdots S_{m-1}} \leftarrow \alpha TH_{m,S_1 \cdots S_{m-1}} + \Omega \cdot (-1)^{1-S_m}$
        end for
    end if
    $\tau_S \leftarrow \tau_S + \Delta_S$
end for
```

Parameters setting. In the experiments, we set the parameters in Algorithm 1 as follows: $\alpha = 0.99$, $\Lambda = 1$, $\Omega = 0.1$. These are the same values as the previous experiment[6]. The signal $s(\tau)$ is represented by an 8-bit integer type: $-128 \leq s(\tau) < 128$.

Convergence of Algorithm 1 based on uniform distribution. This discussion on the convergence concerns only the two-armed bandit, while the random sequences are uniformly distributed and independent each time – something that does not concern chaotic time sequences. We assume that $K = 2$ and the time series used for comparison with thresholds follows a uniform distribution of $[-1/2, 1/2]$ at an arbitrary time. We define the value of threshold TH_1 at the beginning of time step n as $w(n)$. The time evolution of $w(n)$ can be represented as

$$\begin{gathered} w(n+1) = \alpha w(n) + q(n) \\ \text{where} \quad q(n) := \begin{cases} +\Lambda & (\text{if } A(n) = 0, X_0(n) = 1) \\ -\Lambda & (\text{if } A(n) = 1, X_1(n) = 1) \\ +\Omega & (\text{if } A(n) = 1, X_1(n) = 0) \\ -\Omega & (\text{if } A(n) = 0, X_0(n) = 0) \end{cases}. \end{gathered} \tag{8}$$

The expectation of $w(n)$ is represented as follows.

$$\begin{aligned} \mathbb{E}\left[q(n)\right] &= \Lambda \Pr(A(n) = 0, X_0(n) = 1) - \Lambda \Pr(A(n) = 1, X_1(n) = 1) \\ &\quad + \Omega \Pr(A(n) = 1, X_1(n) = 0) - \Omega \Pr(A(n) = 0, X_0(n) = 0) \\ &= \Lambda \mu_0 \Pr(A(n) = 0) - \Lambda \mu_1 \Pr(A(n) = 1) + \Omega(1 - \mu_1)\Pr(A(n) = 1) - \Omega(1 - \mu_0)\Pr(A(n) = 0) \\ &= \{(\Lambda + \Omega)\mu_0 - \Omega\}\Pr(A(n) = 0) - \{(\Lambda + \Omega)\mu_1 - \Omega\}\Pr(A(n) = 1). \end{aligned} \tag{9}$$

Because we assume that $s(t)$ follows a uniform distribution, if $\max\{n\Lambda, n\Omega\} < 1/2$,

$$\begin{aligned}
\Pr(A(n)=0) &= \int \Pr(A(n)=0|w(n)=x)\cdot\Pr(w(n)=x)\mathrm{d}x\\
&= \int\left(\frac{1}{2}+x\right)\Pr(w(n)=x)\mathrm{d}x\\
&= \frac{1}{2}+\mathbb{E}[w(n)],\\
\Pr(A(n)=1) &= \int \Pr(A(n)=1|w(n)=x)\cdot\Pr(w(n)=x)\mathrm{d}x\\
&= \int\left(\frac{1}{2}-x\right)\Pr(w(n)=x)\mathrm{d}x\\
&= \frac{1}{2}-\mathbb{E}[w(n)].
\end{aligned}$$

By Eqs. (8) and (9),

$$\begin{aligned}
\mathbb{E}[w(n+1)] &= \alpha\mathbb{E}[w(n)]+\mathbb{E}\big[q(n)\big]\\
&= \frac{1}{2}(\Lambda+\Omega)(\mu_0-\mu_1)+\{\alpha+(\Lambda+\Omega)(\mu_0+\mu_1)-2\Omega\}\mathbb{E}[w(n)].
\end{aligned} \tag{10}$$

Equation (10) can lead to

$$\mathbb{E}[w(n)] = \frac{P}{1-Q}+Q^{n-1}\left(\mathbb{E}[w(1)]-\frac{P}{1-Q}\right) \tag{11}$$

where

$$P := \frac{1}{2}(\Lambda+\Omega)(\mu_0-\mu_1), \quad Q := \alpha+(\Lambda+\Omega)(\mu_0+\mu_1)-2\Omega.$$

Equation (11) indicates that $\Pr(A(n)=0)$ and $\Pr(A(n)=1)$ converge to a certain value in (0, 1) if $|P/(1-Q)| < 1/2$ and $|Q| < 1$. In this case, the number of selections for each arm linearly increases. Furthermore, if $|P/(1-Q)| \geq 1/2$ or $|Q| \geq 1$, then convergence or divergence occurs at $|\mathbb{E}[w(n)]| \geq 1/2$, which leads to $\Pr(A(n)=1) \approx 1$ or $\Pr(A(n)=0) \approx 1$. In this case, one of the arms will be selected intensively as time passes.

The above discussion shows that the convergence and performance of Algorithm 1 depend on learning rate α, exploration degree (Λ, Ω), and reward environment (μ_0, μ_1).

Details of the proposed method.

Algorithm 2 Proposed method

```
Parameters: $0 < \alpha < 1, \quad \Delta_S > 0, \quad \Delta_L > 0, \quad d \in \mathbb{N}, \quad \beta > 1$
Initialization:
$\tau_S \leftarrow \tau_{init}, \quad t_i \leftarrow 0, \quad r_i \leftarrow 0,$
$\forall m, S_1, \ldots, S_{m-1} : TH_{m,S_1\cdots S_{m-1}} \leftarrow 0, \quad \Lambda_{m,S_1\cdots S_{m-1}} \leftarrow \Lambda_{init}, \quad \Omega_{m,S_1\cdots S_{m-1}} \leftarrow \Omega_{init}$
for $n = 1, \ldots, n_{\max}$ do
    $\tau_L \leftarrow \tau_S$
    $\vdots$
    (Algorithm1: STEP1 – STEP3)
    $\vdots$
    $t_a \leftarrow t_a + 1, \quad r_a \leftarrow r_a + x_a$
    if $n \mod d = 0$ then
        for $m = 1, \ldots, M$ do
            if two confidence intervals are overlapped, then
                $\Lambda_{m,S_1\cdots S_{m-1}} \leftarrow \Lambda_{m,S_1\cdots S_{m-1}}/\beta$
                $\Omega_{m,S_1\cdots S_{m-1}} \leftarrow \Omega_{m,S_1\cdots S_{m-1}}/\beta$
            else
                $\Lambda_{m,S_1\cdots S_{m-1}} \leftarrow \Lambda_{m,S_1\cdots S_{m-1}} \cdot \beta$
                $\Omega_{m,S_1\cdots S_{m-1}} \leftarrow \Omega_{m,S_1\cdots S_{m-1}} \cdot \beta$
            end if
        end for
    end if
    $\tau_S \leftarrow \tau_S + \Delta_S$
end for
```

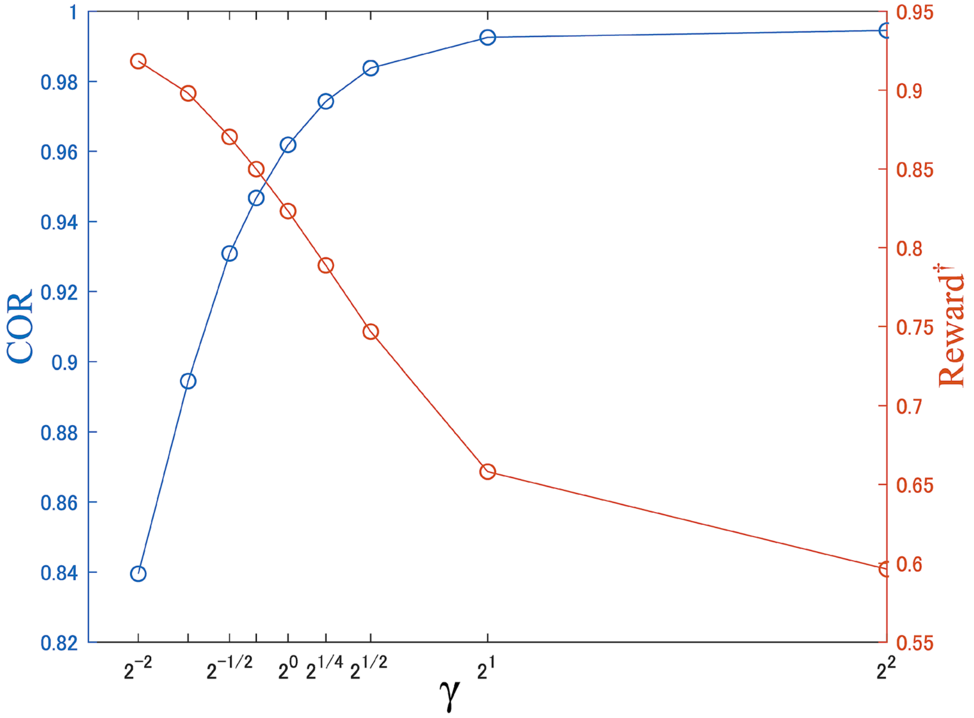

Figure 5. Parameter dependency of γ.

Parameters setting. In the experiments, we set the parameters in Algorithm 1 as follows: $\alpha = 0.99$, $\Lambda_{init} = 1$, $\Omega_{init} = 0.1$, $d = 10$, $\beta = 2$. We also set $\gamma = 1/\sqrt{2}$, which is the scale parameter for the confidence intervals. The signal $s(\tau)$ is converted into an 8-bit integer variable type: $-128 \leq s(\tau) < 128$.

Dependency for scale parameter of confidence intervals. Figure 5 shows the influence of the parameter of confidence intervals. γ is a parameter related to the width of confidence intervals. We can see that the correct order rate becomes higher and the obtained rewards smaller as γ becomes smaller, and vice versa. When $\gamma = \sqrt{2}$, the reward and the correct order rate are relatively high; we use this value for γ in Chaos-CI described in the main text.

Convergence of the proposed method based on uniform distribution. As described above, we have found that the performance of the algorithm proposed is heavily dependent on parameters (Λ, Ω). Therefore, in the proposed method, exploration-degree adjustments based on confidence intervals are added to Algorithm 1: if the exploration itself is not sufficient, then thresholds are set close to 0 and values of (Λ, Ω) decrease, so thresholds are less likely to diverge, which leads to improved accuracy. If exploration is applied sufficiently, then the values of (Λ, Ω) increase, so the thresholds are more likely to diverge, which leads to an intensive selection of a better arm and slow increase of regret.

Data availability

The datasets generated during the current study are available from the corresponding author on reasonable request.

Received: 27 March 2020; Accepted: 8 February 2021
Published online: 24 February 2021

References

1. Lorenz, E. N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963).
2. Fischer, I., Liu, Y. & Davis, P. Synchronization of chaotic semiconductor laser dynamics on subnanosecond time scales and its potential for chaos communication. *Phys. Rev. A* **62**, 011801 (2000).
3. Uchida, A. *Optical Communication with Chaotic Lasers: Applications of Nonlinear Dynamics and Synchronization* (Wiley, Hoboken, 2012).
4. Uchida, A. *et al.* Fast physical random bit generation with chaotic semiconductor lasers. *Nat. Photonics* **2**, 728–732 (2008).
5. Yoshimura, K. *et al.* Secure key distribution using correlated randomness in lasers driven by common random light. *Phys. Rev. Lett.* **108**, 070602 (2012).
6. Naruse, M. *et al.* Scalable photonic reinforcement learning by time-division multiplexing of laser chaos. *Sci. Rep.* **8**, 10890 (2018).
7. Littman, M. L. Reinforcement learning improves behaviour from evaluative feedback. *Nature* **521**, 445–451 (2015).
8. AlphaGo: Mastering the ancient game of Go with Machine Learning. https://ai.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html. Last access: 26 March 2020.

9. Takeuchi, S. *et al.* Dynamic channel selection in wireless communications via a multi-armed bandit algorithm using laser chaos time series. *Sci. Rep.* **10**, 1–7 (2020).
10. Bejarano, O., Knightly, E. W. & Park, M. IEEE 802.11 AC: from channelization to multi-user MIMO. *IEEE Commun. Mag.* **51**, 84–90 (2013).
11. Auer, P., Cesa-Bianchi, N. & Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* **47**, 235–256 (2002).
12. Russo, D., Van Roy, B., Kazerouni, A., Osband, I. & Wen, Z. A tutorial on Thompson sampling. arXiv preprint, arXiv:1707.02038 (2017).
13. Slivkins, A. Introduction to multi-armed bandits. arXiv preprint, arXiv:1904.07272 (2019).
14. Soriano, M. C., Flunkert, V. & Fischer, I. Relation between delayed feedback and delay-coupled systems and its application to chaotic lasers. *Chaos: Interdiscip. J. Nonlinear Sci.* **23**, 043133 (2013).
15. Ohtsubo, J. *Semiconductor Lasers: Stability, Instability and Chaos* (Springer, Berlin, 2012).

Acknowledgements

This work was supported in part by the CREST project (JPMJCR17N2) funded by the Japan Science and Technology Agency and Grants-in-Aid for Scientific Research (A) (JP17H01277) funded by the Japan Society for the Promotion of Science. The authors acknowledge Atsushi Uchida and Kazutaka Kanno for the measurements of laser chaos time series and Satoshi Sunada and Hirokazu Hori for their variable discussions about chaos and order recognition.

Author contributions

M.N. directed the project and N.N. designed the order recognition algorithm and conducted signal processing. N.N., N.C., M.H., and M.N. analyzed the data and N.N. and M.N. wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to N.N. or M.N.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.