XCS-CR for Handling Input, Output, and Reward Noise

Takato Tatsumi
The University of Electro-Communications
Japan
tatsumi@uec.ac.jp

Keiki Takadama
The University of Electro-Communications
Japan
keiki@hc.uec.ac.jp

ABSTRACT

To briefly represent a dataset, it is crucial to find common attributes among the data. Extended learning classifier system (XCS) finds common attributes of multiple data and acquires generalized rules that match multiple data. In real-world problems, it may be challenging to find common attributes due to noise in the data and the inability of XCS to acquire the generalized rules. Considering a classification problem as an example, noise may be included at each input, output, as well as in the evaluation of the output. To tackle this problem, our previous work proposed XCSs that acquire appropriately generalized rules, specifically for a problem with one of the three mentioned type of noises added. In real-world problems, it is difficult to identify the type of noise in advance, which requires an XCS to cope with multiple types of noise. For this issue, this paper proposes an XCS that can handle any noise on the input, output, and evaluation of the output, and aims at investigating the effectiveness of the proposed XCS in the multiplexer problems including any of the three types of noise.

CCS CONCEPTS

• **Computing methodologies → Rule learning**;

KEYWORDS

XCS, accuracy criteria, input noise, output noise, reward noise

ACM Reference Format:
Takato Tatsumi and Keiki Takadama. 2019. XCS-CR for Handling Input, Output, and Reward Noise. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3319619.3326863

1 INTRODUCTION

In data mining, it is crucial to clarify which attributes are essential. Besides, in a classification problem, it is necessary to clarify attributes and their values that affect a classification class. Learning classifier systems (LCSs) [4] finds knowledge in a classification problem using genetic algorithm (GA) [2] and reinforcement learning (RL) [8]. The LCS estimates the class for input and outputs it as well; moreover, it is reward based on the success or failure of the estimated class. Furthermore, LCS treats the identified knowledge with if-then rules called classifiers and can acquire a generalized classifier that matches multiple inputs. The generalized classifiers are expressed by replacing some attributes of the "if" part with a "don't-care" symbol. eXtended LCS (XCS) [16], which is the mainstream of LCS, acquires classifiers that always receive the same reward. Since the (accurate) classifiers acquired by XCS always receive the same reward, the attributes of the "if" part of the classifier replaced by the "don't-care" symbol does not affect the amount of the reward. XCS assumes that if a class is outputted for the same input, the same reward will be acquired. However, in many cases, the assumption cannot be made due to sensor failure, incorrect recording, the uncertainty of the reward function, to mention a few. In data mining not limited to XCS, this problem is addressed by preprocessing. The preprocessing complements data by estimating the reasons for the uncertainty and inconsistency occurring in data. Nevertheless, it is difficult to complement the data correctly by preprocessing since it is extremely costly. Therefore, there is a need for methods to stably acquire knowledge without preprocessing. Additionally, XCS should be able to cope with noise on input, output, and evaluation in order to receive an input, perform an output, and receive an evaluation from the training object. This paper considers a binary classification problem for binary input. In this paper, an input noise inverts some binaries of the input while an output noise inverts the output. A reward noise changes the reward by adding noise to the value based on a Gaussian distribution. XCSs that handle a particular type of noise have been proposed by Lanzi *et al.* [6], Webb *et al.* [15], Lanzi *et al.* [5], and Tatsumi *et al.* [11, 13, 14]. The methods employed by these authors are divided into statistical table-based XCSs and constrained non-statistical table-based XCSs. The statistical table-based XCSs have a mechanism to record the statistical value (mean and standard deviation) of the acquired reward separately from the acquired classifiers; in this case, it is possible to identify the accurate classifiers correctly and stably. Since the statistical table-based XCSs emphasize the stability of learning performance (rule generalization and correct rate), it requires many data to determine the accuracy of the classifiers. On the other hand, the non-statistical table-based XCSs identify accurate classifiers based on the relationships of the acquired classifiers. However, these XCSs have less memory usage compared to the statistical table-based XCSs; besides, there are restrictions on applicable problems associated with the non-statistical table-based XCSs. In this paper, we introduce XCSs that acquire appropriately generalized rules even in variance of reward (XCS-VR)[1] [11] and an XCS based on the mean of reward (XCS-MR) [14], as statistical table-based XCSs, are applicable in environments with input, output, or reward noise added. Therefore, it is

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326863

[1] In [11], this method was called an XCS for Unstable Reward Environment (XCS-URE), but we call the method XCS-VR to clarity its contrast with the following method discussed in this paper.

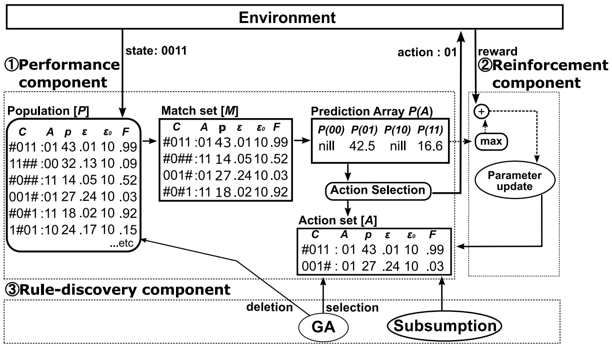


Figure 1: Learning mechanism of XCS (see [10]).

assumed that an XCS based on the range of reward (XCS-RR) [13], which is one of the non-statistical table-based XCSs, is applicable in problems where the range of reward acquired by the accurate classifier is relatively small; that is, the correctness is not reversed by noise. XCS based on the collective weighted reward (XCS-CR) [12], which is a non-statistical table-based XCS, is applicable in problems where the reward is only of the following two kinds: the value is given at the time of a correct answer and at the time of an incorrect answer. In order to expand the application area of non-statistical table-based XCSs, it is necessary to increase the types of noise that could be handled and coped with in the problem of the long input length. This study focuses on creating a new XCS that can handle any noise. Furthermore, in this paper, we propose a non-statistical table-based XCS to acquire suitably generalized classifiers even for learning data that include input, output, or reward uncertainty in binary classification problems. The proposed method is an extension of XCS-CR, such that it can cope even when the reward takes different values.

The rest of this paper is organized as follows: Sections 2, 3, and 4 describe how to judge the rule accuracy in XCS, XCS-CR, and XCS-CR2, respectively. Section 5 modifies the multiplexer problems to add noise as the uncertainty of inputs, outputs, and rewards. Section 6 presents some experiments and results, while Section 7 discusses the results. Finally, Section 8 concludes the paper.

2 ACCURACY-BASED LEARNING CLASSIFIER SYSTEM (XCS)

As shown in Figure 1 above, XCS repeats performance, reinforcement, and rule-discovery components to acquire the classifiers.

2.1 Classifier and its generalization

XCS classifier has the condition (if) part, action (then) part, prediction p, prediction error ϵ (that is, the difference between the prediction and the reward P), fitness F, and numerosity n. XCS acquires knowledge by generating classifiers to fit multiple inputs. When the condition part is represented by a bit string with a fixed length composed of 0 and 1, XCS generalizes the classifiers by using the symbol # representing “don't-care”. For example, “10###” matches eight inputs. The first two bits are emphasized, and the last three bits are ignored. Moreover, XCS aims to cover all inputs with several generalized classifiers.

2.2 The mechanism of XCS

2.2.1 Performance component.

Performance component. XCS selects an action for an identified input and executes it. The algorithm in this component is summarized as follows: (i) Extracts classifiers that match the current input in $[P]$ and stores the extracted classifiers in the match set $[M]$; (ii) XCS predicts the acquisition reward for each action a_i by calculating the prediction array $P(a_i)$ as follows:

$$P(a_i) = \frac{\sum_{cl_k \in [M]|a_i} cl_k.p \times cl_k.F}{\sum_{cl_l \in [M]|a_i} cl_l.F}. \tag{1}$$

(iii) XCS selects an action based on the prediction array and stores the classifiers that have the selected action in the action set $[A]$. The action is generally chosen based on the ϵ-greedy selection [9] or by alternation between random and greedy selections. Then, XCS executes the selected action for the environment and receives the reward P. Next, the reinforcement component is executed followed by the execution of the evolution component after a particular time.

2.2.2 Reinforcement component.

Reinforcement component. This component updates the parameters of the classifiers in $[A]$ as follows: (i) the prediction p is updated based on the acquired reward P;

$$cl.p \leftarrow cl.p + \beta(P - cl.p). \tag{2}$$

The variable β is the learning rate and contributes to the learning speed. (ii) The error ϵ, which is the difference between P and $cl.p$ is updated as follows:

$$cl.\epsilon \leftarrow cl.\epsilon + \beta(|P - cl.p| - cl.\epsilon). \tag{3}$$

However, when the number of updates of the classifier ($cl.exp$) is less than $1/\beta$, Equations (2) and (3) are respectively replaced with Equations (4) and (5) below using moyenne adaptive modifée technique [3]:

$$cl.p \leftarrow cl.p + (P - cl.p)/cl.exp; \tag{4}$$

$$cl.\epsilon \leftarrow cl.\epsilon + (|P - cl.p| - cl.\epsilon)/cl.exp. \tag{5}$$

These operations increase the learning speed at the beginning of learning. (iii) The fitness F is calculated based on the accuracy κ.

$$\kappa(cl) = \begin{cases} 1 & \text{if } \epsilon < \epsilon_0; \\ \alpha \left(\frac{\epsilon}{\epsilon_0}\right)^{-\nu} & \text{otherwise.} \end{cases} \tag{6}$$

In this equation, ϵ_0 ($\epsilon_0 > 0$) is a constant that indicates the accuracy criterion. When the value of $cl.\epsilon$ is less than ϵ_0, the classifier is accurate. The variables α ($0 \leq \alpha \leq 1$) and ν ($\nu > 0$) control the reduction rate of the accuracy. The relative accuracy of the classifier κ' is then calculated as follows:

$$\kappa'(cl) = \frac{\kappa(cl) \times cl.n}{\sum_{x \in [A]} \kappa(x) \times x.n}. \tag{7}$$

(iv) The fitness F is updated as follows:

$$cl.F \leftarrow cl.F + \beta(\kappa'(cl) - cl.F). \tag{8}$$

2.2.3 Rule-discovery component. This component generates new classifiers by GA as follows: (i) Two individual parents individuals are selected from $[A]$; however, the selection probability is based on their fitness ratio. (ii) Next, two different children are generated by crossing the selected parents. The elements of the condition part of these children are mutated with the probability μ. (iii) Then, the children are added in $[P]$. At that time, if the total numerosity n of the classifier in $[P]$ exceeds the parameter N, the classifiers with low fitness are deleted preferentially.

Moreover, XCS generalizes the classifiers and organizes $[P]$. Subsumption is the process of integrating classifiers with a low generality into more generalized (more # in the condition part) classifiers. The classifiers whose experience exp exceeds θ_{sub} and are determined to be accurate ($\kappa(cl) = 1$) can subsume classifiers that have their condition part included in the condition part of the subsuming classifier. Then, the numerosity n of the subsumed classifier is added to the classifier that subsumes.

3 XCS BASED ON COLLECTIVE WEIGHTED REWARD (XCS-CR)

3.1 Architecture of XCS-CR

The classifier of XCS-CR has the condition part, action part, prediction p, prediction error ϵ, fitness F, and numerosity n as the same as those of the classifier of XCS; however, XCS-CR differs from XCS in the following aspects. XCS has one ϵ_0 parameter that is shared by all classifiers, but in XCS-CR, every classifier has its ϵ_0. Additionary, XCS-CR classifiers have the mean of the acquired reward M which is similar to p; however, M is more stable than p. Furthermore, XCS-CR classifier counts the number of times the reward is received ($C_{P=P}$) and the number of estimation for each reward ($E_{P=P}$). In the multiplexer problem, since there are two possible reward values, 0 and 1000, the classifier has four counters $C_{P=0}$, $C_{P=1000}$, $E_{P=0}$, and $E_{P=1000}$.

3.2 Mechanism of XCS-CR

Most parts of the learning mechanism of XCS-CR are the same as those of the mechanism of XCS. Nevertheless, the mechanism of XCS-CR differs mainly from that of XCS in terms of the parameter updating in the reinforcement component and the subsumption condition in the rule-discovery component. Figure 2 shows the learning mechanism of XCS-CR where the parts that distinguished it from that of XCS are blackened.

3.2.1 Reinforcement Component. XCS-CR updates the classifiers in $[A]$ as follows. First, the C_P to be incremented is the one whose value of P is the same as the acquired reward P (*i.e.*, $C_{P=0}$ for reward 0 and $C_{P=1000}$ for reward 1000). Thus,

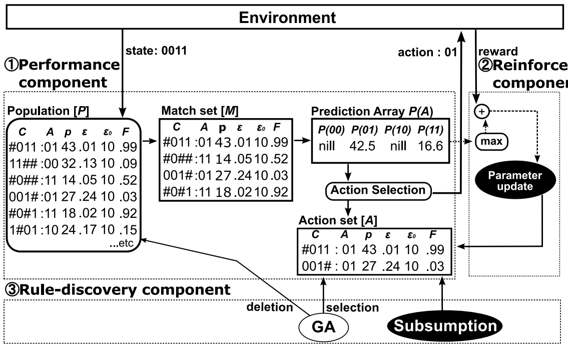
$$cl.C_{P=P} \leftarrow cl.C_{P=P} + 1. \tag{9}$$

Second, the classifier updates the mean of its acquired reward $cl.M$ as follows:

$$cl.M = \frac{1}{cl.exp} \sum_{i=1}^{cl.exp} P_i \tag{10}$$

, where P_i is the reward acquired when the experience exp of the classifier cl is i. Third, the error ϵ of the classifier is updated as shown in Figure 3. Besides, ϵ is calculated as the difference between the most frequently acquired reward $cl.mfr$ and the mean value of the acquired reward. The most frequently acquired reward $cl.mfr$ is the reward value, for which $cl.C_P$ is the maximum. Thus,

$$cl.\epsilon \leftarrow cl.\epsilon + \beta(|cl.mfr - cl.M| - cl.\epsilon). \tag{11}$$



C	A	p	ε	ε₀	F
#011	:01	43	.01	10	.99
11##	:00	32	.13	10	.09
#0##	:11	14	.05	10	.52
001#	:01	27	.24	10	.03
#0#1	:11	18	.02	10	.92
1#01	:10	24	.17	10	.15

C	A	p	ε	ε₀	F
#011	:01	43	.01	10	.99
#0##	:11	14	.05	10	.52
001#	:01	27	.24	10	.03
#0#1	:11	18	.02	10	.92

P(00)	P(01)	P(10)	P(11)
nill	42.5	nill	16.6

C	A	p	ε	ε₀	F
#011	: 01	43	.01	10	.99
001#	: 01	27	.24	10	.03

Figure 2: Learning mechanism of XCS-CR and XCS-CR2 (see [12]).

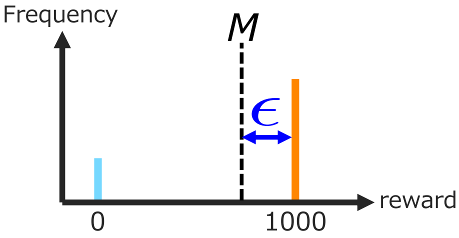
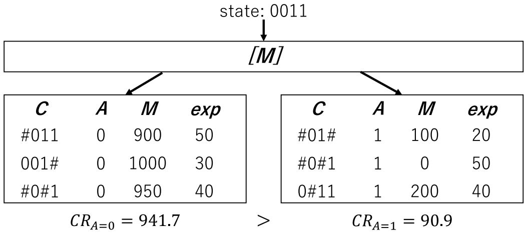


Figure 3: ϵ of XCS-CR and XCS-CR2 (see [12]).



C	A	M	exp
#011	0	900	50
001#	0	1000	30
#0#1	0	950	40

C	A	M	exp
#01#	1	100	20
#0#1	1	0	50
0#11	1	200	40

Figure 4: Calculation of $CR_{A=a}$ of XCS-CR (see [12]).

Fourth, XCS-CR attempts to balance noise effect by estimating the correct action if there is no noise. For each action set $[A]$ XCS-CR calculates a collective reward CR using the equation:

$$CR_{A=a} = \frac{\sum_{cl \in [M]|a} cl.M \times cl.exp}{\sum_{C \in [M]|a} C.exp}. \tag{12}$$

Moreover, XCS-CR assumes the action with the highest CR to be the correct action. The classifier of XCS-CR counts the number of estimations E_P for each reward. However, like C_P, the E_P to be incremented is the one whose value of P is the same as the estimated reward P. The increment is as follows:

$$cl.E_{P=P} \leftarrow cl.E_{P=P} + 1. \tag{13}$$

Figure 4 shows an example of a binary classification problem. XCS-CR consists of $[M]$ with classifiers matching current input "0011". $[M]$ is divided into two collectives based on the action. The classifiers in the left collective have action 0, while the classifiers in the right collective have action 1. Since the M values of the left collective classifiers are larger than those of the right collective classifiers, $CR_{A=0}$ is larger than $CR_{A=1}$. When action 0 is selected, XCS-CR estimates that reward 1000 will be acquired and increments $cl.E_{P=1000}$ by 1. On the other hand, when action 1 is selected, XCS-CR estimates that reward 0 will be acquired and increments $cl.E_{P=0}$ by 1.

Next, XCS-CR updates ϵ_0 of the classifier. The classifier whose estimated reward is always the same, that is, whose E_P is 0 for one reward value, is considered to be accurate. The more # symbols there are in the condition, the more inputs a classifier matches. However, in order to prevent assessment by only some matched inputs, even if the number of experiences exceeds $2^{number\ of\ \#} \times \theta_{RE}$, the classifiers that satisfy the above condition are targeted. Moreover the parameter θ_{RE} is constant. Let ϵ be $Max\epsilon$, the largest ϵ among the classifiers in $[A]$ satisfying the above conditions. Then, XCS-CR updates ϵ_0 as follows:

$$cl.\epsilon_0 \leftarrow cl.\epsilon_0 + \beta(Max\epsilon - cl.\epsilon_0). \tag{14}$$

Finally, XCS-CR updates the fitness F of the classifier executing Equations (6), (7), and (8) the same as XCS does.

XCS-CR decides the accuracy of the classifier based on the estimated value of the acquired reward ($E_{P=P}$). A classifier that has an estimated reward that is always one is considered an accurate classifier, whereas a classifier with two or more estimated reward values considered an inaccurate classifier.

3.2.2 Subsumption condition. Since a classifier with many # symbols subsumes several classifiers, it is necessary to determine the accuracy of such a classifier carefully. XCS has the condition that the number of evaluation times ($cl.exp$) of the subsuming classifier is greater than θ_{sub}; however, in XCS-CR, it is changed to the condition: the number of evaluation times ($cl.exp$) of the subsuming classifiers is greater than $2^{number\ of\ \#} \times \theta_{RE}$. Since the values of θ_{sub} and θ_{RE} are assumed to be the same, the subsumption condition of XCS-CR is more severe than that of XCS. Besides, the other condition $\kappa(cl) = 1$ for XCS-CR is the same as in XCS.

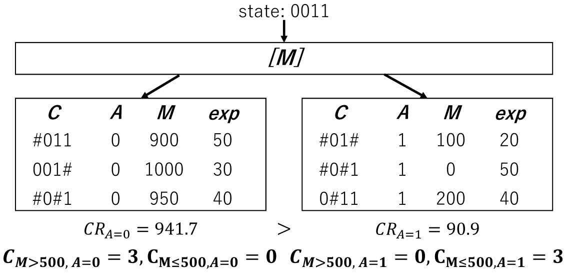
4 EXTENDED XCS BASED ON COLLECTIVE WEIGHTED REWARD (XCS-CR2)

4.1 Architecture

A classifier of XCS-CR2 has the same parameters $\epsilon_0, M, C_{P=P}$, and $E_{P=P}$ as those of a classifier of XCS-CR.

4.2 Mechanism

Most of the mechanism of XCS-CR2 is similar to that of XCS-CR. The mechanism of XCS-CR2 differs mainly from that of XCS-CR in terms of calculation of E_P and determination of $Max\epsilon$ for calculating ϵ_0. This subsection describes the mechanism of XCS-CR2 assuming a binary classification problem in which the reward acquired at the time of correct answer is 1000 when there is no noise.



C	A	M	exp
#011	0	900	50
001#	0	1000	30
#0#1	0	950	40

C	A	M	exp
#01#	1	100	20
#0#1	1	0	50
0#11	1	200	40

$CR_{A=0} = 941.7$ > $CR_{A=1} = 90.9$

$\boldsymbol{C_{M>500,\,A=0} = 3, C_{M\leq500,A=0} = 0 \quad C_{M>500,\,A=1} = 0, C_{M\leq500,A=1} = 3}$

Figure 5: Calculation of $CR_{A=a}, C_{M>500,A=a}$, and $C_{M\leq500,A=a}$ of XCS-CR2.

4.2.1 Calculation of E_P. XCS-CR2 calculates the collective reward *CR before acquiring a reward.* At the same time, it counts the number of macro-classifiers where $cl.M$ is greater than 500 $C_{M>500,A}$ and the number of macro-classifiers where $cl.M$ is less than or equal 500 $C_{M\leq500,A}$ for each output A. Figure 5 shows an example of the calculation of $C_{M>500,A=a}$ and $C_{M\leq500,A=a}$. The left-hand side of the figure is the set of classifiers whose action is 0. Since the set on the left-hand side has three classifiers with $M > 500$ and zero classifiers with $M \leq 500$, $C_{M>500,A=0}$ is three and $C_{M\leq500,A=0}$ is zero. In the set on the right-hand side, $C_{M>500,A=1}$ is zero and $C_{M\leq500,A=1}$ is three, because the Ms of all three classifiers are less than 500.

XCS-CR2 determines the E_P from $CR_{A=a_0}, CR_{A=a_1}$, $C_{M>500,A=a_0}, C_{M>500,A=a_1}, C_{M\leq500,A=a_0}$, and $C_{M\leq500,A=a_1}$, where the actions are either a_0 or a_1.

$$\begin{cases} CR_{A=a_0} > CR_{A=a_1}. \\ C_{M>500,A=a_0} > C_{M\leq500,A=a_0}. \\ C_{M>500,A=a_1} < C_{M\leq500,A=a_1}. \end{cases} \tag{15}$$

When all the above conditions are satisfied, the $cl.E_{P=1000}$ is incremented by one if the action of the classifiers $cl \in [A]$ is zero, whereas the $cl.E_{P=0}$ is incremented by one if the action of the classifiers $cl \in [A]$ is one. Thus,

$$\begin{cases} CR_{A=a_0} \leq CR_{A=a_1}. \\ C_{M>500,A=a_0} < C_{M\leq500,A=a_0}. \\ C_{M>500,A=a_1} > C_{M\leq500,A=a_1}. \end{cases} \tag{16}$$

When all of the above conditions are satisfied, $cl.E_{P=0}$ is incremented by one if the action of the classifiers $cl \in [A]$ is zero, while $cl.E_{P=1000}$ is incremented by one if the action of the classifiers $cl \in [A]$ is one. However, $cl.E_{P=P}$ is not updated if neither of the sets satisfy some of the above conditions. XCS-CR2 repeats this updating process each time $[A]$ is generated.

4.2.2 Determining $Max\epsilon$. First, this subsubsection describes the process of calculation of ϵ. XCS-CR2 updates $C_{P=P}$ based on the collective rewards $CR_{A=a_0}$ and $CR_{A=a_1}$. Besides, the $cl.C_{P=1000}$ is increased by one when the CR of the action of the classifiers in $[A]$ is relatively high, whereas the $cl.C_{P=0}$ is increased by one otherwise. Furthermore, XCS-CR2 calculates the mean of the acquired reward of the classifier $cl.M$, and ϵ as in Equation (11) by using the above $C_{P=P}$.

Next, XCS-CR2 determines $Max\epsilon$ by using ϵ of the classifiers that satisfy the following conditions.

$$\begin{cases} cl.exp > 2^{number\ of\ \#} \times \theta_{RE}. \\ cl.exp > \theta_{LL}. \end{cases} \tag{17}$$

The first condition in Equation (17) is the same as the condition of XCS-CR; however, the second condition indicates the lower limit of the experience of the classifier. The maximum value of ϵ of the classifiers satisfying the above conditions is the $Max\epsilon$. The other updating processes such as subsumption condition are the same as for XCS-CR.

5 PROBLEM DESCRIPTION

This paper simulates real-world problems that have uncertainties as input, output, and reward noise added to the l-Multiplexer problem. This section explains the l-Multiplexer problem as well as the added noise.

5.1 Multiplexer problem

The l-Multiplexer problem is a common benchmark problem of LCS because all inputs are expressed with a few numbers of generalized rules. Moreover, the l-Multiplexer problem classifis l bits input ($b_0 b_1 ... b_{l-1}$) into two classes (0 and 1), where l satisfies $l = k + 2^k$. The first k bits ($b_0 b_1 ... b_{k-1}$) of the input are converted into a decimal number d. The value of the $k+d$-th bit (b_{k+d}) is the correct answer, while the other value of the $k+d$-th bit is the incorrect answer. In the 11-Multiplexer problem where the input length is 11 (*i.e.*, $k = 3$), for example, $b_0 b_1 b_2$ ("011") is converted into $d = 3$ when the input is given as "01100100000," and $b_{3+3} = b_6 = 0$ is the correct answer.

Since the value of the correct answer is determined only from the first k bits ($b_0 b_1 ... b_{k-1}$) and the $(k+d)$-th bit (b_{k+d}), the other bits in this input have no meaning. Furthermore, XCS can generalize the input of the l-Multiplexer problem by replacing the other bits with # as shown in the example below.

(if) 011###0#### (then) 0.

The set of maximally generalized classifiers is called the optimal subset $[O]$. For example, in the 11-Multiplexer problem, $[O]$ consists of 32 generalized classifiers. It is worthy of note that, although the classifiers in $[O]$ can cover all of the input-output space, the subset of classifiers in which the number of the # symbols is the same and the address bits are generalized (*e.g.* (if) #11###0###0 (then) 0) cannot cover part of the input-output space.

5.2 Input, output, and reward noise

This paper added noise to the 11-Multiplexer problem in a similar fashion as in [11–14]. We briefly describe the process in this subsection.

- **Input and output noise**: Since the l-Multiplexer problem is a binary classification problem, even if the input and output change slightly, there will be a change only between the two discrete states, that is, the case where the correct class is changed and where the answer is not changed. However, this paper considers only discrete changes in input and output. The input noise inverts the bits of each input attribute with the probability P_I, while the output noise inverts the action with the probability P_O.
- **Reward noise**: It is known that if the reward changes slightly, the accuracy of the classifiers (of XCS) will be reduced. This paper incorporated continuous noise into the reward. Besides, the rewards are random numbers that follow a Gaussian distribution with mean 0 and variance σ_R^2.

6 EXPERIMENTS

This paper compares the learning performance of XCS, XCS-MR, XCS-RR, XCS-CR, and XCS-CR2 in the 11-Multiplexer environments without noise in the input, output, and reward noise. The magnitude of the input, output, and reward noise is $P_I = 0.05$, $P_O = 0.1$, and $\sigma_R^2 = 200^2$, respectively. In XCS-RR, it is assumed that the range of rewards acquired by the accurate classifiers is narrow; however, the range of acquired reward by the accurate classifiers and that by the inaccurate classifiers are the same (from 0 to 1000). In this paper, we did not apply XCS-RR to environments with input and output noise. Additionally, in XCS-CR, it is assumed that the acquired rewards at the time of accurate and inaccurate answer are exactly one value each, 1000 and 0, respectively. When the reward noise is added, the reward takes various values different from 0 and 1000. However, we did not apply XCS-CR to an environment with the reward noise.

6.1 Evaluation criteria and parameters

In this paper, we adopted the evaluation criteria utilized in [11–14]. The evaluation criteria were performance, population size, and the number of trials of the acquired optimal subset $[O]$, which are explained below.

- **Performance**. This criterion evaluates the rate of outputting the correct answer for the latest 100 inputs without noise; a higher correct rate is preferred to a lower one. In these experiments, the term "performance" corresponds to the correct rate. This criterion is the average value of the 50 trials.
- **Population size**. The population size is the number of macro-classifiers in $[P]$. This criterion evaluates the rule generalization performance. In order to reduce the population size, it is necessary to evaluate the accuracy of the classifiers stably and correctly. A smaller population size is preferred to a larger one because in that case, the required memory size becomes relatively small. In this paper, this criterion is the average value of the 50 trials.
- **Number of trials of the acquired optimal subset** $[O]$. This criterion evaluates the 32 optimal classifiers in the 11-Multiplexer problem, and it consists of two measures. The first measure is the percentage of the trials where all 32 optimal classifiers $[O]$ are in $[P]$ at the end of the trial in the 50 trials. The second measure is the percentage of the trials in which the 32 classifiers, having the largest fitness F among $[P]$, are the 32 optimal classifiers $[O]$ in the 50 trials. In this paper, we call the second measure *top 32*. It is preferred to have more trials to acquire $[O]$.

However, it should be noted that the performance and the population size to be utilized in comparing XCSs is the average of the last

Table 1: Number of trials of the acquired $[0]$ in no noise environment.

Method	Acquisition $[O]$	Acquisition $[O]$ (top 32)
XCS	50 (100%)	50 (100%)
XCS-MR	50 (100%)	50 (100%)
XCS-CR	50 (100%)	50 (100%)
XCS-RR	50 (100%)	50 (100%)
XCS-CR2	50 (100%)	50 (100%)

Table 2: Number of trials of the acquired $[0]$ in an input noise environment.

Method	Acquisition $[O]$	Acquisition $[O]$ (top 32)
XCS	0 (0%)	0 (0%)
XCS-MR	49 (98%)	41 (82%)
XCS-CR	49 (98%)	46 (92%)
XCS-CR2	50 (100%)	50 (100%)

100 iterations of the learning in the 50 trials. The correct rate and population size are significantly verified by the Kruskal-Wallis test and Brunner-Munzel test. Moreover, in this paper, the significance level is 1%.

We implemented XCS base on [1] and adopted the parameters utilized in [7]. We also implemented XCS-MR, XCS-RR, and XCS-CR and adopted their corresponding parameters based on [14], [13], and [12], respectively. There fore, we adopted the following parameters for XCS: $\epsilon_0 = 38.34, \beta = 0.099$, and $\theta_{sub} = 49$. While we adopted the parameters employed in [14], [13], and [12] ($\epsilon_0 = 10, \beta = 0.2, \theta_{sub} = 20$) as the parameter of XCS-MR, XCS-RR, XCS-CR, and XCS-CR2. For the common parameters of XCSs, we have $N = 800, \mu = 0.04, P_{\#} = 0.35, P_{explr} = 1.0, \chi = 0.8, \nu = 5, \theta_{GA} = 25$, and $\theta_{del} = 20$. The value of θ_{LL}, which is newly established in this paper for XCS-CR2, is 25. Besides, each trial ran for 300,000 exploit iterations.

6.2 Results

Figure 6, 7, 8, and 9 show the correct rate and population size of XCS, XCS-MR, XCS-RR, XCS-CR, and XCS-CR2. In these figures, the horizontal axis represents the number of iterations, while the vertical axis indicates the correct rate (in (a) of these figures) or the population size (in (b) of these figures). Square, triangle, circle, cross, and Y marks denote the mean value of XCSs. The top and bottom of the bars respectively indicate the maximum and minimum value of the correct rate or the population size in 50 trials of each method. Table 1, 2, 3, and 4 show the percentage of the trials of acquiring the optimal classifiers subset $[O]$. The left-hand side of the tables shows the percentage when the object is the whole of $[P]$, while the right-hand side shows the percentage when the object is the top 32 of $[P]$. Except for the correct rate in the no noise environment case, the significant differences between the methods are obtained by Kruskal-Wallis test. A * symbol is added whenever there is a significant difference among the methods.

In the no noise environment (see Figure 6), all XCSs can select the correct answer for any inputs. Additionally, all XCSs were able to generalize classifiers; however, the population size of XCS-MR, XCS-CR, and XCS-CR2 is the smallest*. From Table 1, it is obvious that all XCSs can acquire $[O]$ in all the trials. The results in noisy environments (see Figure 7, 8, and 9; Table 2, 3, and 4) are as follows. XCS could not select the correct answer for some inputs, and its correct rate is the lowest in all the noisy environments*. The converged value of the correct rate and the population size of XCS-MR, XCS-RR, XCS-CR, and XCS-CR2 in the noisy environments are almost the same as their values in the no noise environment. Although the convergence speed of the correct rate and the population size of these methods decreased, they could acquire $[O]$ by 300,000 iterations in most trials. However, it was impossible to acquire $[O]$ in some trials, as shown below. It was found that XCS-MR has a wide range of the bars in the input and output noise environments; however, the learning of XCS-MR did not converge at 300,000 iterations. Nevertheless, if more learning data is provided, XCS-MR can acquire $[O]$ in all trials. Some trials of XCS-CR2 in the reward noise environment failed to acquire $[O]$. In one of these trials, it was impossible to acquire $[O]$ even when the learning data was increased. These results revealed that XCS-CR2 could select the correct answer and acquire the optimal classifiers subset $[O]$ in almost all the trials; the same as XCSs in the no noise environment.

Table 3: Number of trials of the acquired $[0]$ in an output noise environment.

Method	Acquisition $[O]$	Acquisition $[O]$ (top 32)
XCS	0 (0%)	0 (0%)
XCS-MR	47 (94%)	30 (60%)
XCS-CR	49 (98%)	46 (92%)
XCS-CR2	48 (96%)	47 (94%)

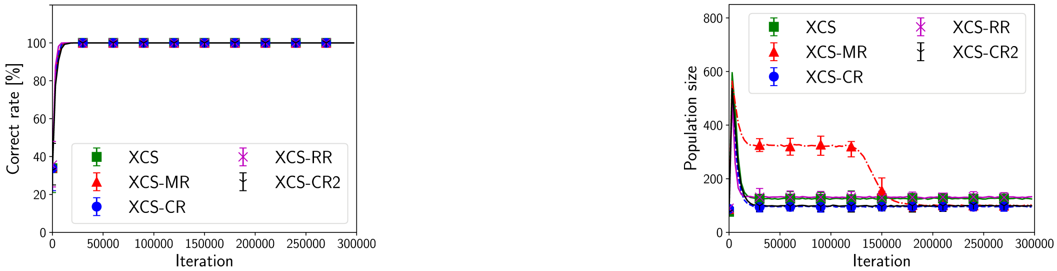
Table 4: Number of trials of the acquired $[0]$ in a reward noise environment.

Method	Acquisition $[O]$	Acquisition $[O]$ (top 32)
XCS	0 (0%)	0 (0%)
XCS-MR	50 (100%)	50 (100%)
XCS-RR	50 (100%)	50 (100%)
XCS-CR2	46 (92%)	44 (88%)

7 DISCUSSIONS

7.1 Introduction of $C_{M>500,A}$ and $C_{M\leq 500,A}$

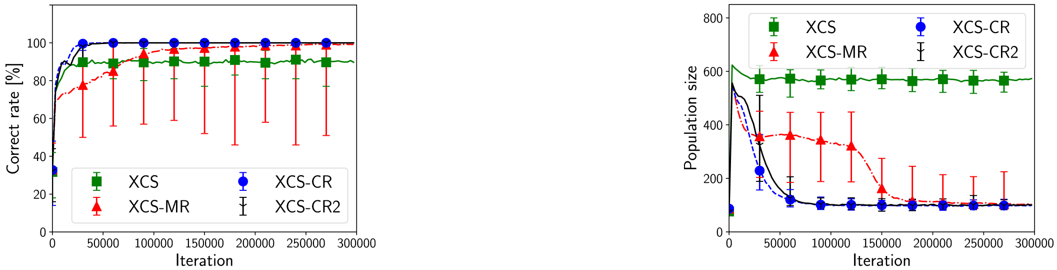
The leaning policy of XCS-CR2 is the same as that of XCS-CR. However, in XCS-CR2, it is difficult to clarify whether the variation of the acquired reward is due to noise or an erroneous (over-) generalization of the information of the classifier to be evaluated. The match set $[M]$ includes classifiers that match inputs from the current input; nevertheless since the classifiers in $[M]$ have a common attribute that match the current input, these classifiers have the most information about the current input. In XCSs, classifiers with a small variation in reward tend to remain in $[P]$, whereas information (*e.g.* M, p, and ϵ) on classifiers with a high experience *exp* is generally, reliable. Consequently, the collective reward *CR*,



(a) Correct rate

(b) Population size

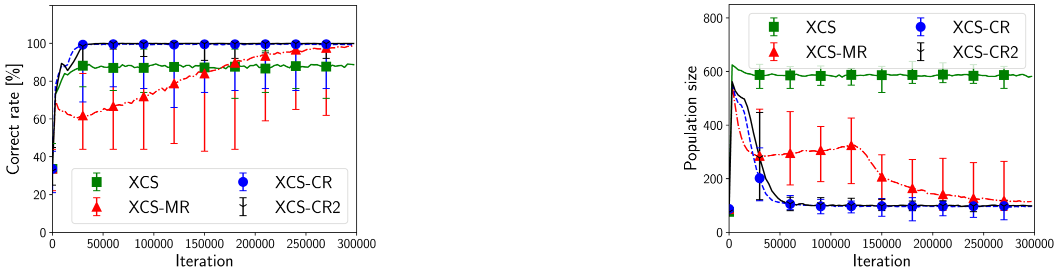
Figure 6: Learning performance in no noise environment.



(a) Correct rate

(b) Population size

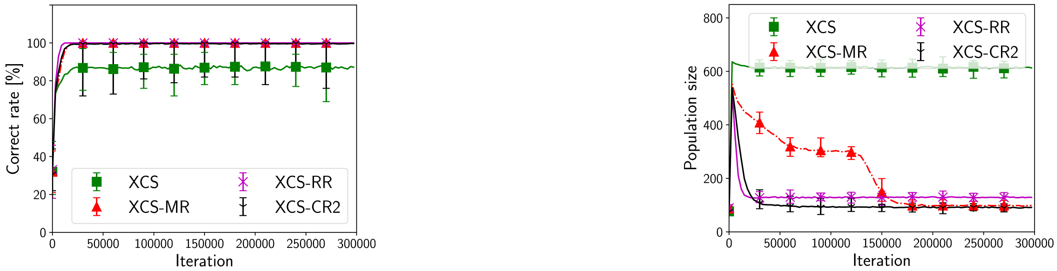
Figure 7: Learning performance in input noise environment.



(a) Correct rate

(b) Population size

Figure 8: Learning performance in output noise environment.



(a) Correct rate

(b) Population size

Figure 9: Learning performance in reward noise environment.

which is the average of M with exp as weight, is an index that is hardly affected by erroneous classifier generalization.

Although the experience exp of the classifiers with high generality (many # symbols) increases rapidly, the collective reward CR is strongly influenced by the classifiers with a large number of exp,

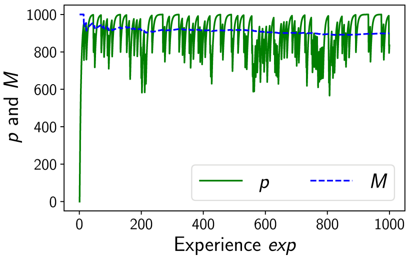


Figure 10: Comparison of p and M.

that is, classifiers that are likely to be over-generalized. XCS-CR2 introduced $C_{M>500,A}$ and $C_{M\leq500,A}$ additionally for the calculation of E_P. However, $C_{M>500,A}$ and $C_{M\leq500,A}$ are indices which are independent of exp. The classifiers with a low exp are relatively accurate since, in this case, they have low generality. Furthermore, XCS-CR2 increases the value of E_P slowly. It is worth noting that on the premise of a binary classification problem in which the reward acquired at the time of correct answer is 1000 and that acquired at the time of incorrect answer is 0, 500, which is the midpoint between 0 and 1000 is set as the threshold.

The disadvantage of the slowly increasing E_P is that the E_P of the over-generalized classifiers to be determined as inaccurate are difficult to increases. In addition, the over-generalized classifiers have the mean of the acquired reward M to be close to 500 and also have a large number of exp. Moreover, since it is difficult to satisfy the Equations (15) and (16), E_P hardly increases. Both $E_{P=0}$ and $E_{P=1000}$ of the inaccurate classifier should be greater than 0. However, since either $E_{P=0}$ or $E_{P=1000}$ may remain 0, the classifier is regarded as an accurate classifier candidate, and ϵ of the classifier is a candidate for the $Max\epsilon$. Furthermore, since the ϵ of the over-generalized classifier is larger than that of the accurate classifiers, the ϵ of the letter is adopted as the $Max\epsilon$; consequently, ϵ_0 is enlarged. The accurate classifiers are subsumed by the over-generalized classifier that is determined as accurate. This explains why XCS-CR2 could not select the correct answer in some trials in the reward noise environment.

7.2 The subsumption condition of XCS-CR2

Equation (17) shows the subsumption condition in XCS-CR2. In the case where the subsumption condition is only the first condition common to XCS-CR and XCS-CR2, the low generality classifiers are used to calculate the $Max\epsilon$ even if the experience exp is relatively small. For example, the classifiers that are not generalized at all are used for the calculation of $Max\epsilon$, even if exp is zero. Furthermore, suppose the classifier is accurate; if exp is small, the possibility that the number of acquisitions of rewards that take different values from the original value due to the influence of the noise is larger than the number of acquisitions of the original reward is sufficiently high. In order to correctly grasp the influence of noise, a certain number of evaluations are required to be carried out. XCS-CR2 set a lower limit θ_{LL} to exp.

7.3 Adoption of mean M for calculation of collective reward CR

XCS-CR and XCS-CR2 adopt the mean value M of acquired reward instead of the prediction p for calculating the collective reward CR. In order to acquire correctly generalized classifiers in environments to which noise is added, it is critical to understand the influence of noise on the reward accurately. Furthermore, XCS-CR and XCS-CR2 adopt an index that can stably and correctly determine the accuracy of classifiers rather than the speed with which it is determined. Figure 10 shows the prediction p and the mean M of an accurate classifier in the output noise ($P_O = 0.1$) environment. The horizontal axis represents the experience, while the vertical axis represents p and M. The value of p is calculated by Equation (2), and it ranges between 600 and 1000. From Equation (2), one observes that the weight for the p of recently acquired rewards is relatively large. If the classifier acquires reward 1000, p is close to 1000, while if the classifier acquires reward 0, p will diminish. On the other hand, the mean M treats all acquired rewards with the same weight. As the number of acquired rewards used for the calculation of M increases, the value of M converges. In this example, M converges to approximately 900. Based on these observations, XCS-CR and XCS-CR2 adopt the mean value M of acquired reward instead of the prediction p for calculating the collective reward CR.

Classifiers whose outputs are different from the correct and incorrect answer, even if one of the inputs matched, are inaccurate. As the input length increases, classifiers that match more inputs are generated. However, in environments with long input length, the difference in the M is relatively small, making it difficult to distinguish between accurate and inaccurate classifiers. In such environments, the p that sensitively reflects the difference in the acquired rewards may be more appropriate than M. Hence; there is a need to explore more efficient indicators further.

8 CONCLUSION

This paper proposes a new XCS, XCS-CR2 (extended XCS-CR) that can accurately generalize classifiers in input, output, and reward noise environments. XCS-CR2 is a method extended to make XCS-CR learnable even in reward noise environments by using the average of the reward acquired at the time of correct and incorrect answer as the threshold. Moreover, XCS-CR2 can acquire generalized classifiers without prior knowledge of the type of added noise.

In this paper, we have experimentally shown that XCS-CR2 can acquire $[O]$ of the 11-Multiplexer problem in environments where one of input, output, and reward noise was added. XCS-MR and XCS-CR2 are methods that can acquire $[O]$ in all noise environments. However, since XCS-CR2 can generalize classifiers faster than XCS-MR, it can be applied to problems with less learning data. Notwithstanding, XCS-MR should be used if the reliability of learning performance is desired, because XCS-CR2 may not acquire $[O]$ in several trials.

Critical research in the field needs to be pursued in the future aimed at the following: (1) improvement of the estimation acquiring reward mechanism to apply to long input length problem; (2) adaptation to the multi-class classification problems; and (3) adaptation to environments with multiple type noise and real-world problems.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP17J03593.

REFERENCES

[1] M. V. Butz and S. W. Wilson. 2002. An algorithmic description of XCS. *Soft Computing* 6, 3-4 (2002), 144–153.

[2] D. E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). Addison-Wesley Longman Publishing Co., Inc.

[3] G.Venturini. 1994. Adaptation in Dynamic Environments through a Minimal Probability of Exploration. In *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior.* 371–381.

[4] J. H. Holland. 1986. Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. *Machine learning* (1986), 593–623.

[5] P. L. Lanzi and M. Colombetti. 1999. An Extension to the XCS Classifier System for Stochastic Environments. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99).* 353–360.

[6] P. L. Lanzi and S. W. Wilson. 2000. Toward Optimal Classifier System Performance in Non-Markov Environments. *Evol. Comput.* 8, 4 (Dec. 2000), 393–418.

[7] M. Nakata, W. Browne, T. Hamagami, and K. Takadama. 2017. Theoretical XCS Parameter Settings of Learning Accurate Classifiers. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17).* ACM, 473–480.

[8] R. S. Sutton. 1988. Learning to Predict by the Methods of Temporal Differences. *Machine Learning* 3, 1 (1988), 9–44.

[9] R. S. Sutton and A. G. Barto. 2011. Reinforcement learning: An introduction. (2011).

[10] T. Tatsumi, T. Komine, M. Nakata, H. Sato, T. Kovacs, and K. Takadama. 2016. Variance-based Learning Classifier System without Convergence of Reward Estimation. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (GECCO '16 Companion).* ACM, 67–68.

[11] T. Tatsumi, T. Komine, H. Sato, and K. Takadama. 2015. Handling different level of unstable reward environment through an estimation of reward distribution in XCS. In *2015 IEEE Congress on Evolutionary Computation (CEC).* 2973–2980.

[12] T. Tatsumi, T. Kovacs, and K. Takadama. 2018. XCS-CR: Determining Accuracy of Classifier by Its Collective Reward in Action Set Toward Environment with Action Noise. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18).* ACM, New York, NY, USA, 1457–1464.

[13] T. Tatsumi, H. Sato, and K. Takadama. 2017. Automatic Adjustment of Selection Pressure Based on Range of Reward in Learning Classifier System. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17).* ACM, New York, NY, USA, 505–512.

[14] T. Tatsumi, H. Sato, and K. Takadama. 2017. Learning Classifier System Based on Mean of Reward. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 21, 5 (2017), 895–906.

[15] A. Webb, E. Hart, P. Ross, and A. Lawson. 2003. *Controlling a Simulated Khepera with an XCS Classifier System with Memory.* Springer Berlin Heidelberg, Berlin, Heidelberg, 885–892.

[16] S. W. Wilson. 1995. Classifier Fitness Based on Accuracy. *Evol. Comput.* 3, 2 (June 1995), 149–175.