Theoretical Analysis of Accuracy-Based Fitness on Learning Classifier Systems

RUI SUGAWARA AND MASAYA NAKATA, (Member, IEEE)
Graduate School of Engineering Science, Yokohama National University, Yokohama 240-8501, Japan
Corresponding author: Rui Sugawara (sugawara-rui-sg@ynu.jp)

This work was supported by the JSPS KAKENHI under Grant 20H04254.

ABSTRACT Like most evolutionary algorithms, accuracy-based learning classifier systems (XCSs) use a fitness metric to recognize the superiority of rules, under a principle that a higher-quality rule has a higher fitness. However, XCS must learn the fitness values under a reinforcement learning scheme. This introduces uncertainty and asynchrony to the fitness estimation while no theoretical work formally guarantees that such a basic principle would hold. The goal of this paper is to complement this fundamental lack in the reliability of XCS by mathematically analyzing its fitness-update scheme. Our main assumption is that the fitness is updated with an absolute accuracy instead of its relative accuracy to sidestep unpredictable dynamics of XCS. Our theoretical conclusion is that the superiority of rules can be correctly recognized through the accuracy-based fitness under finite update times. We further show that recognizing the superiority among low-quality rules is a costly procedure that increases the number of necessary rule-trainings. This drawback indicates that XCS may struggle to identify good parent rules at early generations, degrading the efficiency of evolutionary propagation.

INDEX TERMS Accuracy-based fitness, learning classifier system, theoretical analysis.

I. INTRODUCTION

Learning Classifier Systems (LCSs) [1] are a paradigm of evolutionary rule-based learning approaches, aiming to produce accurate and maximally general rules that capture the complexity behind a given problem [2]. LCSs seek to build an ensemble model of simple condition-action rules, wherein each rule is responsible for a sub-input space divided optimally with evolutionary algorithms. During the last decade, LCSs have been variously extended in response to modern interest in the machine learning field, for example, ensemble modeling of neural networks [3]–[6], knowledge extraction with deep neural network models [7]–[9], and automation of LCSs [10].

Accuracy-based LCSs (XCSs) [11] are the most common basis of modern LCSs designed for various problem domains, such as classification, recognition, and online-control [12]–[14]. XCS optimizes rules with a genetic algorithm (GA) and evaluates their quality through a reinforcement learning scheme. Its main feature is the utilization of an accuracy-based fitness, in which the rule-quality is considered as an estimation accuracy of expected rewards to be received by its rule. In early works, the impact of accuracy-based fitness has been actively studied, as fitness design is crucial to boosting evolutionary propagation. For instance, Kovacs’s optimality hypothesis suggested that XCS can develop minimal representations of optimal solutions [15]–[17]. Butz discussed how specific XCS components act to produce the optimal rules while utilizing the fitness [18], [19]. For the sake of simplicity, hereinafter, the accuracy-based fitness is denoted as fitness, if not stated otherwise.

Like most evolutionary algorithms, the rule-discovery process of XCS, that is, GA, optimizes rules relying on a basic principle that a higher-quality rule has a higher fitness. However, XCS must estimate the fitness from a limited resource of reward signals [20], causing uncertainty and asynchrony in the fitness estimation. Specifically, the fitness of rules is asynchronously trained because each rule covers a different sub-input space [21]. Furthermore, the fitness value may differ, even for the same rule-quality, owing to some probabilistic factors, for example, the input distribution. Thus, we can consider that XCS seeks to solve an uncertain (i.e., noisy) and asynchronous optimization problem, where the fitness may be unreliable. XCS may thus fail to correctly recognize the superiority of rules.

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.
In this regard, the collected insights are briefly summarized below.

- **Handling uncertainty.** Using a small learning rate, $\beta$, can improve the estimation accuracy of rule-variables, including fitness [22], and thus, can relax the uncertainty in the fitness estimation [23]. Although this strategy deteriorates the convergence speed of the rule-variables, it is effective, especially for large-scale and/or noisy problems, as shown in [24]. For this, several works have attempted to adaptively tune the learning rate during a run [25]–[27]. The brief function theory [28] and an estimation technique for reward distributions [29] are integrated into XCS to manage the uncertainty in classification problems.

- **Handling asynchrony.** A population size bound is theoretically discussed in [22], [30], [31], given that high-quality rules can have reproductive events before they are deleted. This condition, known as the reproductive opportunity bound, is a reasonable explanation to manage the asynchrony in fitness estimation. Frequency tuning of GA [32], [33] and a theoretical condition to safely delete rules [21] have also been considered as additional perspectives.

- **Other perspectives.** Certain works investigated the affinity of the fitness with possible options of the XCS framework. The impact of fitness-sharing [34]–[36], niche GAs [37], and parent selection methods [38], [39] have been analyzed. For instance, tournament selection tends to improve the stability of XCS performance [40].

However, some fundamental insights related to the fitness have remained theoretically unclear so far. Specifically, it has not been proved yet whether XCS can correctly recognize the superiority of any rules through their fitness, and how many times each rule should be trained to represent its quality. The former issue is related to the understanding of XCS stability against the uncertainty in fitness estimation. The latter corresponds to quantifying a negative impact of the asynchrony in the fitness-update scheme. Considering these issues, this work presents a theoretical analysis of the fitness-update scheme. Our goal is to theoretically conclude that XCS can correctly recognize the superiority of any rules through their fitness under finite training times (i.e., a feasible condition). Thus, our main contribution is to provide the first proof that the basic principle, that is, a higher-quality rule has a higher fitness, holds, complementing a fundamental lack in the reliability of XCS. We therefore demonstrate that, the XCS framework is designed to manage uncertainty and asynchrony in optimizing rules online under assumptions.

The mathematical analysis of the fitness value is generally non-trivial because of the complexity in deriving its estimation equation. In fact, the fitness is updated with four variables: a prediction, a prediction error, and absolute/relative accuracies (see Section II for more detail). The absolute accuracy is determined with a discontinuous function, and estimations of the prediction and prediction error are further required to analyze the fitness. Furthermore, the relative accuracy is used to realize the fitness sharing approach, but this has an unpredictable impact on the fitness estimation.

To relax the above difficulties, we consider recent theoretical insights and assumptions. Specifically, the estimation equations of prediction and prediction error have been derived in the learning optimality theory [24]; we show that the discontinuity of absolute accuracy can be reasonably decomposed by utilizing these equations. Because of assumptions used in this theory, our analysis is conducted for classification tasks. Furthermore, we assume that the fitness is updated with the absolute accuracy instead of its relative accuracy to avoid its unpredictable property. While this assumption creates a gap in the standard XCS framework, it is a necessary condition to proceed with our analysis. Such simplifications of the fitness calculation have been designed for some LCS variants [34], [41]–[43].

This paper is organized as follows. Section II introduces the XCS framework and derivations of the learning optimality theory. Section III describes our analysis and introduces assumptions and definitions. In Section IV, we further analyze the negative impact of asynchrony in the fitness-update scheme. Conclusions and future work are then summarized in Section V. It is worth noting that, on the classification tasks, the estimation accuracy of expected rewards corresponds to a classification accuracy, and XCS systematically defines the worst rule-quality as having a 50% classification accuracy. This is known as the fitness dilemma, and a bilateral accuracy has been introduced to sidestep this issue [30]. However, this paper focuses on the standard XCS and follows the definition of worst quality.

## II. XCS AND LEARNING OPTIMALITY THEORY

We describe the XCS framework for classification tasks with binary inputs, wherein the rule-condition is represented with a ternary alphabet coding [11]. Subsequently, the learning optimality theory is introduced. Note that this theory does not suppose any particular coding, and thus, can be satisfied for XCSR [44] and XCSI [45], designed for real-valued and integer inputs, respectively.

### A. XCS FRAMEWORK

To apply XCS to the classification tasks, a binary reward scheme is frequently employed [11], [30], in which the system receives $r_{\text{max}}$ if it predicts the correct class for a given input, otherwise $r_{\text{min}}$. This paper sets $r_{\text{min}} = 0$ and $r_{\text{max}} = 1000$.

1) **RULE FORMAT**

A rule has a condition $C$ and an action $A$. The condition determines a set of inputs (i.e., a sub-input space) that the corresponding rule can use. The action corresponds to a predicted class. The condition is represented as $C = [c_1, \ldots, c_d]$ and $c_i \in \{0, 1, \#\}$ for a binary input, where a don’t care symbol # can be any value. Specifically, a rule can be matched to an input $x = [x_1, \ldots, x_d]$ if $c_i$ is equal to $x_i$ or # for all
$i \in \{1, 2, \ldots, d\}$, which is simply denoted as $x \in C$ in this paper. The rule has the following main rule-variables:

- $p_n$ a prediction, which is an estimated reward calculated from given rewards;
- $\epsilon_n$ a prediction error, which estimates an absolute error of $p_n$ to given rewards;
- $\kappa_n$ an absolute accuracy, which is calculated from $\epsilon_n$;
- $F_n$ an accuracy-based fitness, which is calculated from a relative accuracy based on $\kappa_n$;
- $num$ a numerosity, which is the number of subsumed rules.

The above notations of $p_n$, $\epsilon_n$, $\kappa_n$, and $F_n$ indicate values which have been updated $n$ times. Their initial values, except for $\kappa_n$, are denoted as $p_1$, $\epsilon_1$, and $F_1$ for $n = 0$. $\kappa_n$ is directly calculated from $\epsilon_n$. Note that each rule has its own update time $n$, and thus, the rule-variables are asynchronously updated. All rules are contained in a population $\mathcal{P}$ with the maximum size $N$.

2) MECHANISM

To begin with, $\mathcal{P}$ is initialized as an empty set. Given an input $x = [x_1, \ldots, x_d]$, XCS builds a match set $\mathcal{M}$ containing rules matched to $x$, that is, $\mathcal{M} = \{cl \in \mathcal{P} \mid x \in cl.C\}$. If $\mathcal{M}$ does not cover all possible classes, the covering operator is activated to produce rules which have missing actions and conditions matched to $x$. Their initial values in this case are sufficiently small [20], for example, $p_1$, $\epsilon_1$, $F_1 = 0.01$. XCS executes an action $a^*$ selected with a defined strategy and a reward $r$ is then returned to the system.

Next, XCS builds an action set $\mathcal{A}$, which consists of rules having $a^*$, that is, $\mathcal{A} = \{cl \in \mathcal{M} \mid cl.A = a^*\}$, and it updates the rule-variables of rules in $\mathcal{A}$. First, the update time $n$ is increased by one. Then, $p_n$, $\epsilon_n$, and $\kappa_n$ are updated as follows:

$$p_n = p_{n-1} + \beta(r - p_{n-1}),$$

(1)

$$\epsilon_n = \epsilon_{n-1} + \beta(|r - p_{n-1}| - \epsilon_{n-1}),$$

(2)

$$\kappa_n = \begin{cases} 1 & \text{if } \epsilon_n < \epsilon_0, \\ \alpha \left(\frac{\epsilon_n}{\epsilon_0}\right)^{-\nu} & \text{otherwise}, \end{cases}$$

(3)

where $\beta$ is a learning rate; $\epsilon_0$ is an error tolerance; and $\alpha$, $\nu$ control the exponential curve of $\kappa_n$. The fitness, $F_n$, is updated as

$$F_n = F_{n-1} + \beta \left(\kappa'_n - F_{n-1}\right)$$

(4)

where $\kappa'_n$ is a relative accuracy, given by

$$\kappa'_n = \frac{\kappa_n \times \text{num}}{\sum_{cl \in \mathcal{A}} cl.k_n \times cl.num}. $$

(5)

Finally, GA is applied to $\mathcal{A}$ depending on a hyper-parameter $\theta_{GA}$ (see [20] for more details). XCS selects two parent rules from $\mathcal{A}$ and duplicates them as two offspring rules. Then, it applies the crossover and mutation operators to the offspring rules with probabilities $\chi$, $\mu$, respectively. If the crossover is applied, the rule-variables of the offspring rules are set to the average values of the parent rules. Typically, the initial fitness is forcedly reduced with a fitness reduction rate $\delta_F \in (0, 1]$. The offspring rules are inserted to $\mathcal{P}$ and two rules are deleted from $\mathcal{P}$ if its size is greater than $N$.

The subsumption operator may be applied to the rules in $\mathcal{A}$ after updating them and to the offspring rules after GA. A rule $cl_1$ can be subsumed by a more general rule $cl_2$ if $cl_2$ is maximally accurate and sufficiently updated, which is formalized as the following condition:

$$\epsilon_n < \epsilon_0 \lor n > \theta_{sub}$$

(6)

If $cl_1$ is subsumed by $cl_2$, $cl_2.num$ is increased by $cl_1.num$ and $cl_1$ is eliminated from $\mathcal{P}$.

After completing the above procedure, the next input is sent to XCS.

B. LEARNING OPTIMALITY THEORY

Suppose the classification tasks, the maximally accurate rules defined in (6) must have an optimum classification accuracy. However, this optimality has not been proved theoretically. The learning optimality theory guarantees that this optimality holds under some assumptions while identifying a theoretical setting of $\beta$ and $\epsilon_0$. For readability, according to [24], we denote the maximally accurate rules and other rules as accurate and inaccurate ones, respectively.

1) DEFINITION AND GOAL

Let $\phi \in [0, 1]$ be a true classification accuracy of a rule, which is the rate of correct classifications over all inputs that it matches. An accurate rule is defined as having $\phi^* \in \Phi^* = [0, 1]$ because it can correctly predict a given reward. An inaccurate rule is defined as having $\phi' \in \Phi' = [1 - \phi_{\text{max}}', \phi_{\text{max}}']$, where a boundary of the inaccurate rules can be controlled with $\phi_{\text{max}}' \in [0.5, 1)$.\(^1\)

Let $\epsilon_n(\phi)$ be a prediction error of a rule having $\phi$. Note that $\epsilon_n(\phi)$ may change even for the same value of $\phi$ because it depends on $p_1$ and $\epsilon_1$. Furthermore, let max $\epsilon_n(\Phi)$ and min $\epsilon_n(\Phi)$ be the maximum and minimum prediction errors of any rules having $\phi \in \Phi$, that is,

$$\max \epsilon_n(\Phi) := \max_{p_1, \epsilon_1, \phi \in \Phi} \epsilon_n(\phi)$$

(7)

$$\min \epsilon_n(\Phi) := \min_{p_1, \epsilon_1, \phi \in \Phi} \epsilon_n(\phi).$$

(8)

From (6), to correctly distinguish any accurate rules from any inaccurate ones, the following condition must be satisfied;

$$\max \epsilon_n(\Phi^*) < \epsilon_0 \leq \min \epsilon_n(\Phi') \quad \text{s.t. } n^*, n' > \theta_{sub}. $$

(9)

The goal of the theory is to derive the optimal setting of $\beta$, $\epsilon_0$ that satisfies the above condition. Note that in the theory, $\theta_{sub}$ remains as a hyper-parameter, which defines the minimum update time required to identify the accurate rules. This paper uses an ideal setting of $\theta_{sub}$ given that any inaccurate rule receives at least one pair of

\(^1\) $\Phi'$ can be also defined as $\Phi' = \{\phi'_{\text{min}}, \phi'_{\text{max}}\}$. However, $\phi'_{\text{min}}$ is identified as $1 - \phi_{\text{max}}'$ when $\phi_{\text{max}}'$ is defined once, because $\epsilon_n$ follows a symmetric function [24].
\( r_{\text{max}}/r_{\text{min}} \) rewards [24], that is,
\[
\theta_{\text{sub}} = \left[ \frac{1}{1 - \phi'_{\text{max}}} \right] - 1.
\tag{10}
\]

2) DERIVATION

The theory derives estimation equations of \( p_n, \epsilon_n \), and then identifies max \( \epsilon_n(\Phi^*) \) and min \( \epsilon_n(\Phi') \).

Let \( E_r(\phi) \) be an expected reward which a rule having \( \phi \) receives at any update time \( n \), given by
\[
E_r(\phi) = r_{\text{max}}\phi + r_{\text{min}}(1-\phi) = 1000\phi.
\tag{11}
\]

By recursively expanding (1) with \( r \approx E_r(\phi) \), an estimation equation of \( p_n \) for \( \phi \), i.e., \( p_n(\phi) \), is derived as
\[
p_n(\phi) \simeq 1000\phi - (1000\phi - p_1)(1-\beta)^n.
\tag{12}
\]

Subsequently, let \( E_{\epsilon,n}(\phi) \) be an expected value of \( |r - p_{n-1}| \) of (2). Because \( p_{n-1} \) is now replaced with \( p_{n-1}(\phi), E_{\epsilon,n}(\phi) \) is determined as
\[
E_{\epsilon,n}(\phi) = |r_{\text{max}} - p_{n-1}|\phi + |r_{\text{min}} - p_{n-1}|(1-\phi) \\
\simeq 1000\phi + (1 - 2\beta)p_{n-1}(\phi).
\tag{13}
\]

By recursively expanding (2) with \( |r - p_{n-1}| \simeq E_{\epsilon,n}(\phi) \), an estimation equation of \( \epsilon_n(\phi) \) is identified as
\[
\epsilon_n(\phi) \simeq \epsilon_i(1-\beta)^n + \epsilon_\infty(\phi)(1-(1-\beta)^n) \\
- n(1-2\beta)(1000\phi - p_1)\beta(1-\beta)^n-1.
\tag{14}
\]

where \( \epsilon_\infty(\phi) \) is given by
\[
\epsilon_\infty(\phi) = 2000(\phi - \phi^2).
\tag{15}
\]

For \( n \to \infty, \epsilon_n(\phi) \) eventually converges to \( \epsilon_\infty(\phi) \). Consequently, the following lemma holds.

Lemma 1: Assume that two rules have \( \phi \) and \( 1 - \phi \), respectively, wherein \( \phi \in [0.5, 1] \) and their prediction errors can be approximated by (14). When their prediction errors have been updated for the same \( n \) update time, the following two equalities hold:
\[
\begin{align*}
\max \epsilon_n(\phi) &= \max \epsilon_i(1-\phi), \\
\min \epsilon_n(\phi) &= \min \epsilon_i(1-\phi),
\end{align*}
\tag{16}
\]

where \( \max \epsilon_n(\phi) \) and \( \min \epsilon_n(\phi) \) are prediction errors maximized and minimized in terms of the initial values \( i.e., p_1, \epsilon_i \), respectively.

Proof: From (14), \( \epsilon_i(\phi) \) and \( \epsilon_i(1-\phi) \) are maximized with \( p_1, \epsilon_i \) = \( [0, 1000] \) and \( [1000, 1000] \) for any value of \( \beta \), respectively, which are formalized as the same equation as
\[
\max \epsilon_n(\phi) = \max \epsilon_i(1-\phi) \\
= \epsilon_\infty(\phi) - \epsilon_\infty(\phi)(1 - \beta + n\beta)(1 - \beta)^n-1 \\
+ 1000n(1-\beta)\beta(1-\beta)^n-1.
\tag{17}
\]

Similarly, those are minimized with \( p_1, \epsilon_i \) = \( [1000, 0] \) and \( [0, 0] \), respectively, which follow the same equation as
\[
\min \epsilon_n(\phi) = \min \epsilon_i(1-\phi) \\
= \epsilon_\infty(\phi) - \epsilon_\infty(\phi)(1 - \beta + n\beta)(1 - \beta)^n-1
\]

\[
+ 1000n(1 - \phi)\beta(1 - \beta)^{n-1}.
\tag{18}
\]

Note that (17) and (18) are available for \( \phi \geq 0.5 \).
Recall that \( \phi^* \) is either 0 or 1, from (17) with \( \phi = 1 \), max \( \epsilon_n(\Phi^*) \) is immediately derived as
\[
\max \epsilon_n(\Phi^*) = 1000(1 - \beta + n^*\beta)(1 - \beta)^{n^*}-1.
\tag{19}
\]

where \( \epsilon_\infty(\phi^*) = 0. \) For inaccurate rules, \( \min \epsilon_n(\Phi') \) is obtained by minimizing (18) for \( \phi' \). Specifically, it is minimized when \( \phi' = \phi'_{\text{max}} \) or \( 1 - \phi'_{\text{max}} \). Thus, any possible value of \( \epsilon_n(\phi') \) is always greater than or equal to \( \min \epsilon_n(\Phi') \), given by
\[
\begin{align*}
\min \epsilon_n(\Phi') &= \epsilon_\infty(\phi') \\
- \epsilon_\infty(\phi'_{\text{max}})(1 - \beta + n'\beta)(1 - \beta)^{n'-1} \\
&+ 1000n'\beta(1 - \phi'_{\text{max}})(1 - \beta)^{n'-1}
\end{align*}
\tag{20}
\]

From the above derivations, (9) always holds, as proven in the following theorem.

Theorem 1: Assume that \( \phi'_{\text{max}} \) and \( \theta_{\text{sub}} \) are given in advance and a prediction error of any rules can be approximated by (14). Let \( \beta \) be a solution of the following boundary equation at \( n^*, n' = \theta_{\text{sub}} \):
\[
\max \epsilon_{\theta_{\text{sub}}}(\Phi^*) = \min \epsilon_{\theta_{\text{sub}}}(\Phi').
\tag{21}
\]

and \( \epsilon_0 \) be its boundary value calculated with the solution \( \beta \), that is,
\[
\epsilon_0 = \max \epsilon_{\theta_{\text{sub}}}(\Phi^*).
\tag{22}
\]

Then, with the above setting of \( \beta, \epsilon_0 \), the condition (9) holds, that is, XCS always correctly distinguishes the accurate rules from the inaccurate ones when they have been updated more than \( \theta_{\text{sub}} \) times.

Proof: From (19) and (20), max \( \epsilon_n(\Phi^*) \) > min \( \epsilon_n(\Phi') \) holds for \( n^*, n' = 0 \), whereas max \( \epsilon_n(\Phi^*) < \min \epsilon_n(\Phi') \) eventually holds because \( \epsilon_\infty(\phi'_{\text{max}}) \geq 0 \). Thus, there is the intersection point of max \( \epsilon_n(\Phi^*) \) and \( \min \epsilon_n(\Phi') \), and thus, the solution \( \beta \) of (21) always exists. Although (21) cannot be solved algebraically for \( \beta \), the solution \( \beta \) can be obtained by, for example, the Newton’s method.

Here, \( \max \epsilon_n(\Phi^*) \) and \( \min \epsilon_n(\Phi') \) are monotonically decreasing and increasing for \( n^*, n' > 0 \), respectively (see [24] for the formal proof). Because \( \theta_{\text{sub}} > 0 \), max \( \epsilon_n(\Phi^*) \) and \( \min \epsilon_n(\Phi') \) are always smaller and greater than the boundary value \( \epsilon_0 \) for \( n^*, n' > \theta_{\text{sub}} \) respectively. That is, (9) always holds when the prediction and prediction error of any rules are updated with the solution \( \beta \).

Accordingly, the above theorem reveals the theoretical settings of \( \beta, \epsilon_0 \).

III. ANALYSIS

Our analysis inherits theoretical insights proven in the learning optimality theory. We start by introducing assumptions, preparation, and goal. Subsequently, the detailed procedures are described. Note that our conclusion is graphically explained in Section III-C.
A. PREPARATION

1) ASSUMPTIONS

We use the following assumptions.

1) The prediction error can be approximated by (14), and thus, Lemma 1 and Theorem 1 hold.
2) The fitness $F_n$ is updated with the absolute accuracy $\kappa_n$ instead of its relative accuracy $\kappa_n'$.
3) The initial fitness $F_1$ is always sampled from a numerical range of $\kappa_n$.

The second assumption is necessary to advance our analysis because analyzing the relative accuracy is non-trivial due to an unpredictable value of $num$. Note that $\beta$ and $\epsilon_0$ are fixed values not dependent on our analysis, as their values are uniquely determined when $\phi_{\text{max}}$ is defined. The third assumption is used to identify the minimum value of $F_n$ (see Section III-A2 for more details).

2) BASIC INSIGHT

Let $\kappa_n(\phi)$ be an absolute accuracy of a rule having $\phi$ at update time $n$, and $F_n(\phi)$ be its rule-fitness. From the first assumption and (3), $\kappa_n(\phi)$ is rewritten as

$$
\kappa_n(\phi) \simeq \begin{cases} 
1 & \text{if } \epsilon_n(\phi) < \epsilon_0, \\
\alpha \left( \frac{\epsilon_n(\phi)}{\epsilon_0} \right)^{-\nu} & \text{otherwise},
\end{cases}
$$

(23)

where its numerical range can be identified as

$$
\kappa_n(\phi) \in \left[ \alpha \left( \frac{1000}{\epsilon_0} \right)^{-\nu}, 1 \right].
$$

(24)

This is because $\epsilon_n(\phi) \in [0, 1000]$. From the second assumption, the update equation of $F_n(\phi)$ is rewritten as

$$
F_n(\phi) = F_{n-1}(\phi) + \beta (\kappa_n(\phi) - F_{n-1}(\phi))
$$

$$
= F_1(1 - \beta)^n + \beta \sum_{i=1}^{n} \kappa_i(\phi)(1 - \beta)^{n-i},
$$

(25)

which can be proved by mathematical induction [24]. The above equation, that is, the Widrow-Hoff learning rule, is designed not to exceed the numerical range of $\kappa_n(\phi)$ if $F_1$ is sampled from its range. Thus, with our third assumption, the minimum value of $F_n$ appears, which corresponds to that of $\kappa_n(\phi)$. Note that the minimum value of $F_n$ is otherwise indeterminate because $F_1$ may be reduced recursively with a reduction bias of $\delta_F$.

Accordingly, the following lemma holds.

**Lemma 2: Assume two rules having $\phi$ and $1 - \phi$, respectively, wherein $\phi \in [0.5, 1]$. When their fitness has been updated for $n$ update time, the following equalities hold:**

$$
\begin{align*}
\max F_n(\phi) &= \max F_n(1 - \phi), \\
\min F_n(\phi) &= \min F_n(1 - \phi).
\end{align*}
$$

(26)

**Proof:**

From (25), $\max F_n(\phi)$ and $\min F_n(\phi)$ are obtained by maximizing and minimizing $F_1$ and $\kappa_i(\phi)$, respectively, as follows:

$$
\max F_n(\phi) = (1 - \beta)^n + \beta \sum_{i=1}^{n} \max \kappa_i(\phi)(1 - \beta)^{n-i},
$$

(27)

$$
\min F_n(\phi) = \alpha \left( \frac{1000}{\epsilon_0} \right)^{-\nu} (1 - \beta)^n + \beta \sum_{i=1}^{n} \min \kappa_i(\phi)(1 - \beta)^{n-i}.
$$

(28)

From (23), it is obvious that $\min \kappa_i(\phi)$ and $\max \kappa_i(\phi)$ are obtained when $\epsilon_i(\phi)$ are equal to $\max \epsilon_i(\phi)$ and $\min \epsilon_i(\phi)$, respectively. From Lemma 1, $\min \kappa_n(\phi) = \min \kappa_n(1 - \phi)$ and $\max \kappa_n(\phi) = \max \kappa_n(1 - \phi)$ holds, and thus, (26) also holds. $\square$

Importantly, Lemma 2 reveals that we only need to consider $\phi \geq 0.5$ in analyzing the fitness. This theoretically valid simplification will be the basis of our analysis.

3) GOAL

Suppose two rules have $\phi_1, \phi_2 \in [0.5, 1]$, respectively, where $\phi_1 > \phi_2$ holds. Thus, the rule with $\phi_1$ is better than the other with $\phi_2$. The optimality of the fitness estimation argued here is that the rule having $\phi_1$ has a higher fitness than that of the rule having $\phi_2$ at a finite update time. Because their fitness changes depend on the variables of $F_1$ and $\epsilon_n(\phi)$, this optimality must hold for any possible values of these variables, which is strictly formalized as

$$
\min F_n(\phi_1) > \max F_n(\phi_2) \quad \text{s.t. } n_1, n_2 < \infty.
$$

(29)

The above condition is general because it is valid for both accurate and inaccurate rules. However, $\max F_n(\phi)$ and $\min F_n(\phi)$ are differently formalized depending on whether $\phi$ belongs to $\Phi^a$ or $\Phi^i$ owing to the discontinuity in calculating $\kappa_n(\phi)$. Thus, to proceed with the analysis, (29) is expressly divided into the following two specific cases:

- **Superiority of accurate rules against inaccurate rules.**

  The first case is defined such that any fitness of any accurate rules is greater than that of any inaccurate ones, which is conditioned as

  $$
  \min F_n(\Phi^a) > \max F_n(\Phi^i) \quad \text{s.t. } n^a, n^i < \infty,
  $$

  (30)

  where $\min F_n(\Phi^a)$ and $\max F_n(\Phi^i)$ are the minimum and maximum fitness of any accurate and inaccurate rules, respectively.

- **Superiority among inaccurate rules.** Assume two inaccurate rules having $\phi_1', \phi_2' \in \Phi^i$ and $\phi_1' > \phi_2'$. Then, any fitness of the rule with $\phi_1'$ is higher than that of the other with $\phi_2'$, that is,

  $$
  \min F_n(\phi_1') > \max F_n(\phi_2') \quad \text{s.t. } n_1', n_2' < \infty.
  $$

  (31)

When the above two conditions are satisfied, (29) also holds.
B. DERIVATION

To begin with, we reveal the monotonicity of \( F_n(\phi) \) and max \( F_n(\phi) \) for update time \( n \).

**Lemma 3:** Assume that \( \kappa_n(\phi) \) is an increasing function for any update time \( n \). Then, \( \min F_n(\phi) \) is also an increasing function for \( n \).

**Proof:** We prove that \( \min F_n(\phi) \leq \min F_{n+1}(\phi) \) holds. This inequality is rewritten as

\[
\min F_n(\phi) \leq \min \kappa_{n+1}(\phi), \tag{32}
\]

because, from (25) and (28), \( F_{n+1}(\phi) \) is formalized as

\[
\min F_{n+1}(\phi) = \min F_n(\phi) + \beta (\min \kappa_{n+1}(\phi) - \min F_n(\phi)). \tag{33}
\]

Because \( \min \kappa_n(\phi) \) is an increasing function, \( \min \kappa(\phi) \) involved in \( \min F_n(\phi) \), i.e., (28), is smaller than or equal to \( \min \kappa_n(\phi) \), for any index \( i \in \{1, 2, \ldots, n\} \). Accordingly, \( \min F_n(\phi) \) never exceeds \( \min \kappa_n(\phi) \). Accordingly,

\[
\min F_n(\phi) \leq \min \kappa_n(\phi) \leq \min \kappa_{n+1}(\phi). \tag{34}
\]

**Lemma 4:** Assume that \( \max \kappa_n(\phi) \) is a decreasing function for any update time \( n \). Then, \( \max F_n(\phi) \) is also a decreasing function for \( n \).

**Proof:** The proof is given by reversing the inequality signs of Lemma 3. \( \square \)

1) SUPERIORITY OF ACCURATE RULES AGAINST INACCURATE RULES

Here we show that (30) holds, concluding that any accurate rules eventually have fitness higher than that of any inaccurate rules. We start by deriving specific equations of \( \min F_n(\Phi^*) \) and \( \max F_n(\Phi^*) \).

As revealed in (28), \( \min F_n(\Phi^*) \) is determined with the minimum absolute accuracy of any possible accurate rules, denoted as \( \min \kappa_n(\Phi^*) \). From Theorem 1, \( \min \kappa_n(\Phi^*) \) is identified as

\[
\min \kappa_n(\Phi^*) = \begin{cases} 1 & \text{if } n* > \theta_{sub}, \\ \alpha \left( \frac{\max \epsilon_n(\Phi^*)}{\epsilon_0} \right)^{-\nu} & \text{otherwise}, \end{cases} \tag{35}
\]

because \( \max \epsilon_n(\Phi^*) \) is smaller than \( \epsilon_0 \) if and only if \( n* > \theta_{sub} \). Thus, from (28), \( \min F_n(\Phi^*) \) is determined as

\[
\min F_n(\Phi^*) = \alpha \left( \frac{1000}{\epsilon_0} \right)^{-\nu} (1 - \beta)^n + \beta \sum_{i=1}^{\theta_{sub}} \alpha \left( \frac{\max \epsilon_i(\Phi^*)}{\epsilon_0} \right)^{-\nu} (1 - \beta)^{n^*-i} + \beta \sum_{i=\theta_{sub}+1}^{n*} (1 - \beta)^{n^*-i}, \tag{36}
\]

where its converged value for \( n^* \to \infty \), that is, \( \min F_{\infty}(\Phi^*) \), is determined as

\[
\min F_{\infty}(\Phi^*) = \lim_{n^* \to \infty} \beta \sum_{i=\theta_{sub}+1}^{n^*} (1 - \beta)^{n^*-i} = 1. \tag{37}
\]

Meanwhile, \( \max F_n(\Phi') \) is obtained with \( \max \kappa_n(\Phi') \), that is, the maximum absolute accuracy of any inaccurate rules. For any update time \( n' \), \( \max \kappa_n(\Phi') \) is given by

\[
\max \kappa_n(\Phi') = \begin{cases} 1 & \text{if } n' < \theta_{sub}, \\ \alpha \left( \frac{\max \epsilon_n(\Phi')}{\epsilon_0} \right)^{-\nu} & \text{otherwise}, \end{cases} \tag{38}
\]

because \( \min \epsilon_n(\Phi') < \epsilon_0 \) holds if and only if \( n' < \theta_{sub} \), as proven in Theorem 1. Thus, \( \max F_n(\Phi') \) is specified as

\[
\max F_n(\Phi') = (1 - \beta)^{n'} + \beta \sum_{i=1}^{\theta_{sub}} (1 - \beta)^{n'-i} + \beta \sum_{i=\theta_{sub}}^{n'} \alpha \left( \frac{\min \epsilon_i(\Phi')}{\epsilon_0} \right)^{-\nu} (1 - \beta)^{n'-i}. \tag{39}
\]

To identify \( \max F_{\infty}(\Phi') \) for \( n' \to \infty \), we assume that \( \min \epsilon(\Phi') \) of (39) is approximated by its fully-converged value, that is, \( \epsilon_{\infty}(\Phi'_{max}) \). Consequently, \( \max F_{\infty}(\Phi') \) is identified as

\[
\max F_{\infty}(\Phi') = \lim_{n' \to \infty} \alpha \left( \frac{\epsilon_{\infty}(\Phi'_{max})}{\epsilon_0} \right)^{-\nu} = \alpha \left( \frac{\epsilon_{\infty}(\Phi'_{max})}{\epsilon_0} \right)^{-\nu}. \tag{40}
\]

Note that \( \max F_{\infty}(\Phi') \) is always smaller than \( \min F_{\infty}(\Phi^*) \).

From the above discussion, \( \min F_{\infty}(\Phi^*) \) and \( \max F_{\infty}(\Phi') \) now include only one variable, that is, \( n^*, n' \), respectively.

**Theorem 2:** Assume that rule-variables are updated with the theoretical setting of \( \beta \) and \( \epsilon_0 \) adjusted for \( \Phi'_{max} \). Let \( \theta_F \) be an update time that first holds the following inequality:

\[
\min F_{\theta_F}(\Phi^*) > \max F_{\theta_F}(\Phi'). \tag{41}
\]

Then, any possible accurate rules have a higher fitness than that of any inaccurate rules when they have been updated for \( n^*, n' \geq \theta_F \).

**Proof:** A proof can be obtained with a procedure similar to Theorem 1. First, \( \min F_{\theta_F}(\Phi^*) < \max F_{\theta_F}(\Phi^*) \) holds for \( n^*, n' = 0 \), whereas \( \min F_{\infty}(\Phi^*) \) is greater than \( \max F_{\infty}(\Phi') \). Thus, there is a finite solution \( \theta_F \) that satisfies (41).

Herein, \( \min \kappa_n(\Phi^*) \) and \( \max \kappa_n(\Phi') \) are increasing and decreasing functions, respectively. From Lemmas 3 and 4, \( \min F_n(\Phi^*) \) and \( \max F_n(\Phi') \) are also increasing and decreasing functions, respectively. Accordingly, \( \min F_n(\Phi^*) \) is always greater than \( \max F_n(\Phi') \) for \( n^*, n' \geq \theta_F \) when (41) is once satisfied at \( \theta_F \) update time. \( \square \)
Note that $\theta_F$ cannot be described algebraically but can be obtained by, for example, a bisection method.

2) SUPERIORITY AMONG INACCURATE RULES

The goal condition (31) can be proved with almost the same procedure as in Section III-B1. We determine estimation equations of $\min F_n'(\phi'_1)$ and $\max F_n'(\phi'_2)$.

From (28), for any update time $n'_1$, $\min \kappa_{n'_1}(\phi'_1)$ involved in $\min F_{n'_1}(\phi'_1)$ is obtained as

$$\min \kappa_{n'_1}(\phi'_1) = \alpha \left( \frac{\max \epsilon_{n'_1}(\phi'_1)}{\epsilon_0} \right)^{-v}$$

(42)

because $\max \epsilon_{n'_1}(\phi'_1)$ is always greater than $\epsilon_0$, as proven in Theorem 1. Thus, $\min F_{n'_1}(\phi'_1)$ is identified as

$$\min F_{n'_1}(\phi'_1) = \alpha \left( \frac{1000}{\epsilon_0} \right)^{-v} (1 - \beta)^{n'_1} + \beta \sum_{i=1}^{n'_1} \alpha \left( \frac{\epsilon_i(\phi'_1)}{\epsilon_0} \right)^{-v} (1 - \beta)^{n'_1-i}.$$  

(43)

Because $\min \kappa_{n'_1}(\phi'_1)$ is an increasing function, $\min F_{n'_1}(\phi'_1)$ is also increasing for $n'_1$, as proven in Lemma 3.

Next, we consider $\max F_{n'_2}(\phi'_2)$. Here, $\min \kappa_{n'_2}(\phi'_2)$ may exceed $\epsilon_0$ earlier than $\theta_{sub}$ update times because it may be larger than $\min \epsilon(\Phi')$. Let $\theta$ be an update time when $\min \kappa_{n'_2}(\phi'_2)$ is reaching $\epsilon_0$. Thus, $\max \kappa_{n'_2}(\phi'_2)$ is 1 only for $n'_2 < \theta$. Consequently, $\max F_{n'_2}(\phi'_2)$ is derived as

$$\max F_{n'_2}(\phi'_2) = (1 - \beta)^{n'_2} + \beta \sum_{i=1}^{\theta-1} (1 - \beta)^{n'_2-i} + \beta \sum_{i=\theta}^{n'_2} \alpha \left( \frac{\min \epsilon_i(\phi'_2)}{\epsilon_0} \right)^{-v} (1 - \beta)^{n'_2-i}.$$  

(44)

which is a decreasing function, as discussed in Theorem 2. Note that $\theta$ can be calculated with Newton’s method by solving the equation of $\min \epsilon(\phi'_2) = \epsilon_0$.

Herein, with the same procedure as in (40), the converged values of $\min F_{n'_1}(\phi'_1)$ and $\max F_{n'_2}(\phi'_2)$ can be obtained as $\alpha \left( \epsilon_\infty(\phi'_1)/\epsilon_0 \right)^{-v}$ and $\alpha \left( \epsilon_\infty(\phi'_2)/\epsilon_0 \right)^{-v}$, respectively. Because $\phi'_1 > \phi'_2$, the following inequality holds:

$$\min F_{\infty}(\phi'_1) > \max F_{\infty}(\phi'_2).$$

(45)

From the above discussion, we can prove the goal condition (31) with almost the same procedure as in Theorem 2, as summarized in the formal proof below.

**Theorem 3:** Assume two rules having $\phi'_1$ and $\phi'_2$, respectively, and their rule-variables updated with the theoretical setting of $\beta$ and $\epsilon_0$ adjusted for $\phi'_{\max}$. Let $\theta_F$ be an update time that first holds the following inequality:

$$\min F_{\theta_F}(\phi'_1) > \max F_{\theta_F}(\phi'_2).$$

(46)

Then, the rule having $\phi'_1$ has a higher fitness than that of the other rule having $\phi'_2$ when they have been updated for any update time $n'_1, n'_2 \geq \theta_F$.

**Proof:** From (43) and (44), $\min F_{\theta_F}(\phi'_1) < \max F_{\theta_F}(\phi'_2)$ holds for $n'_1, n'_2 \geq 0$, whereas $\min F_{\theta_F}(\phi'_1)$ is eventually greater than $\max F_{\theta_F}(\phi'_2)$, as shown in (45). Thus, there is a finite solution $\theta_F$ of (46). Because $\min F_{n'_1}(\phi'_1)$ and $\max F_{n'_2}(\phi'_2)$ are increasing and decreasing functions, respectively, $\min F_{n'_1}(\phi'_1)$ is always greater than $\max F_{n'_2}(\phi'_2)$ when the equality of (46) is once satisfied at $\theta_F$ update times.

**C. GRAPHICAL EXPLANATION**

To graphically demonstrate the conclusions of our analysis, we use a particular theoretical setting of Case 1, which is derived for $\phi'_{\max} = 0.9$ (see Table 1). Specifically, with $\phi'_{\max} = 0.9$, we can get $\theta_{sub} = 9$ from (10). By solving the boundary equation of (21), its solution is obtained as $\beta = 0.319347$, and thus, $\epsilon_0 = 163.763$. Note that $\alpha$ and $v$ are set to their default values [20].

We first revisit the condition (30), that is, any accurate rules will have a higher fitness than that of any inaccurate rules.

Figure 1 (a) shows the theoretical curves of $\max \epsilon(\Phi')$ and $\min \epsilon(\Phi')$, which explains the conclusion of the learning optimality theory. As proved in Theorem 1, $\max \epsilon(\Phi')$ is equal to $\min \epsilon(\Phi')$ at $\theta_{sub}$ update times; since then, those are smaller and greater than $\epsilon_0$, respectively, and therefore, (9) has been satisfied. Correspondingly, as shown in Figure 1 (b), $\min \kappa_{n'}(\Phi^*)$ and $\max \kappa_{n'}(\Phi^*)$ are equal to 1 for $n' > \theta_{sub}$ and $n' < \theta_{sub}$, respectively.

Figure 1 (c) shows $\min F_{\theta_{sub}}(\Phi^*)$ and $\max F_{\theta_{sub}}(\Phi^*)$. Note that $\theta_F = 11$ is mathematically determined for Case 1 by solving (41). From this figure, the following insights can be confirmed.

- As validated in Lemmas 3 and 4, $\min F_{n'}(\Phi^*)$ and $\max F_{n'}(\Phi^*)$ are increasing and decreasing functions, respectively;
- $\min F_{n'}(\Phi^*)$ converges to 1, whereas $\max F_{n'}(\Phi^*)$ converges to $\alpha(\epsilon_\infty(\phi'_{\max})/\epsilon_0)^{-v} = 0.062333$.
- As proved in Theorem 2, $\min F_{n'}(\Phi^*)$ is greater than $\max F_{n'}(\Phi^*)$ when they have been updated for more than or equal to $\theta_F$ update times.

Furthermore, Figure 2 shows the plots of $\max F_{n'}(\phi')$ with different values of $\phi' \in \Phi^*$. Although $\Phi^*$ can be any set sampled from $[0.1, 0.9]$, we set $\Phi^* = \{0.1, 0.2, \cdots, 0.9\}$

| Parameter | Case 1 | Case 2 | Case 3 |
|-----------|--------|--------|--------|
| $\phi'_{\max}$ | 0.9 | 0.99 | 0.999 |
| $\theta_{sub}$ | 9 | 99 | 99 |
| $\beta$ | 0.319347 | 0.0577475 | 0.00844112 |
| $\epsilon_0$ | 163.763 | 19.580406 | 1.995800 |
| $\Phi'$ | [0.1, 0.9] | [0.01, 0.99] | [0.001, 0.999] |
| $\Phi^*$ | \{0, 1\} | \{0, 1\} | \{0, 1\} |
| $\alpha$ | 0.1 | 0.1 | 0.1 |
| $\nu$ | 5 | 5 | 5 |
for demonstration purposes. Note that \( \max F_n(\psi') = 0.9 \) is equal to \( \max F_n(\Phi') \). As proved in Lemma 2, for \( \psi' \geq 0.5 \), \( \max F_n(\psi') \) and \( \max F_n(1 - \psi') \) follow the same values. Moreover, it is confirmed that \( \max F_n(\Phi') \) is always greater than that with the other values of \( \psi' \). Thus, any accurate rules will have a higher fitness than that of any inaccurate rules when they have been updated for \( n \geq \theta_F \).

Next, to demonstrate that the condition (31) holds, we set \( \psi'_1 = 0.9 \) and \( \psi'_2 = 0.7 \) as an example, where \( \theta = 1 \) and \( \theta_F = 13 \) are obtained. Figure 3 (a) shows the plots of...
max $\epsilon_{\phi}(\phi_1')$ and min $\epsilon_{\phi}(\phi_2')$. As noted in (42), max $\epsilon_{\phi}(\phi_1')$ is always greater than $\epsilon_0$ and min $\epsilon_{\phi}(\phi_2')$ exceeds $\epsilon_0$ earlier than $\theta_{\text{sub}}$. Consequently, min $\kappa_{\phi}(\phi_2')$ is always smaller than 1, as shown in Figure 3 (b). Figure 3 (c) shows the plots of min $F_{\theta_0}(\phi_1')$ and max $F_{\theta_0}(\phi_2')$. min $F_{\theta_0}(\phi_1')$ is always greater than max $F_{\theta_0}(\phi_2')$ when they have been updated for $n \geq \theta_F$, which is consistent with Theorem 3.

### IV. DISCUSSION

It is a known fact that XCS, as well as other LCSs, suffers to solve high-dimensional problems [46], [47]. For instance, XCS has completely failed to improve the rule-quality on the 135-bit multiplexer problem [24], [46]. A common insight to this fact is that discovering better rules is hindered because of a huge search space of rules. Thus far, certain works have improved the rule-discovery process of XCS [48]–[52]. However, the rule-discovery process of XCS is still expected to improve the rule-quality in a stepwise manner, say, from $\phi = 0.5$ to 0.55, and then, to 0.6. Thus, a question is why such a stepwise propagation may be hindered in the high-dimensional problems. In this section, by utilizing our theoretical results, we present an insight in this regard.

The boundary update time, $\theta_F$, indicates a necessary number of rule-trainings to recognize the superiority between two rules having $\phi_1$ and $\phi_2$. Importantly, calculating $\theta_F$ corresponds to quantifying a negative impact of the asynchrony in the fitness estimation. We empirically investigate this impact by analyzing $\theta_F$ adapted for different pairs of $\phi_1$ and $\phi_2$. Specifically, we set $\phi_1' = 0.9, 0.99, 0.999$, respectively, and their parameter settings are summarized in Case1, 2, 3 of Table 1. Note that the theoretical value of $\beta$ decreases when $\phi_1'$ increases. The dependency of $\theta_F$ on $\beta$ will be also discussed below.

Figure 4 (a) shows specific values of $\theta_F$ for Case1, where $\phi_1$ and $\phi_2$ are sampled from $[0, 0.1, 0.15, \cdots, 0.85, 0.9, 1]$. The vertical and horizontal axes indicate the values of $\phi_1$ and $\phi_2$, respectively. Each cell is colored according to the obtained $\theta_F$. Note that the blank (i.e., white) cell indicates that no valid value of $\theta_F$ exists because a pair of $\phi_1$ and $\phi_2$ is out of the definition (see Section III-A3). As explained in the previous section, accurate rules will have fitness values higher than that of any possible inaccurate rules after $n \geq \theta_F = 11$. Recalling $\theta_{\text{sub}} = 9$ at Case1, there is almost no significant difference between $\theta_F$ and $\theta_{\text{sub}}$. However, $\theta_F$ gradually increases when $\phi_1$ and $\phi_2$ are decreasing towards 0.5. For instance, $\theta_F$ increases to 31 when $\phi_1 = 0.55$ and $\phi_2 = 0.5$. In this sense, additional 20 update times are required to recognize the superiority of rules having $\phi_1 = 0.55$, compared to recognizing that of the accurate rules. The above trend is further highlighted when $\beta$ is increasing. Figures 4 (b) and (c) show $\theta_F$ for Case2, 3, where $\theta_F$ for the pair of $\phi_1 = 0.55$ and $\phi_2 = 0.5$ increases to 361 and 3881, respectively.

In summary, recognizing the superiority among low-quality rules is costly, as it requires more update times. As aforementioned, the rule-discovery process is expected to improve the rule-quality in a stepwise manner. However, our observations reveal that such a stepwise propagation is systematically
hindered because the number of rule-trainings required to recognize the superiority among low-quality rules increases. Consequently, XCS may struggle to identify good parent rules from early generations. Moreover, this inefficiency is further highlighted when decreasing $\beta$ to a sufficiently small value. As mentioned in Section I, using a small value of $\beta$ is a reasonable strategy to manage the uncertainty in the fitness estimation, but this conflicts with the relaxation of the negative impact of the asynchrony. In this sense, our result indicates that recognizing the superiority of low-quality rules with as few trainings as possible is crucial to improving the scalability of XCS.

V. CONCLUSION

In this paper, we theoretically provided the first proof that XCS can correctly recognize the superiority of any rules through an accuracy-based fitness under finite update times. Although this insight is limited to the assumption that the fitness is updated according to its absolute accuracy, it provides an insight into the rationality of the XCS’s fitness update scheme. However, our analysis further revealed that additional training of rules is required to recognize the superiority among low-quality rules. This fact indicates that XCS suffers in identifying good parent rules in GA, and thus, it hinders an evolutionary propagation at early generations, where the population is fulfilled mostly with baseline rules.

Additionally, we provided a mathematical form for analyzing the XCS’s fitness-update scheme while linking it to the recent theoretical insight. Presented equations and proofs should be available in XCS variants, e.g., XCSR and XCSI, if they employ the same learning scheme as in XCS. Thus, further advanced insights are expected by investigating the impact of relative accuracy and fitness reduction. Moreover, we focused on the classification task with a binary reward scheme because of the assumption of the learning optimality theory, however, a recent work has attempted to extend it to a multiple reward scheme used in regression tasks [53]. Our analysis should be available in this scheme with this extended theory, as it can utilize the theory without any modifications.

As a different direction, our result points out the importance of reducing $\theta_F$. Considering that the fitness value depends on hyper-parameters of $\alpha$ and $\nu$, there may exist an adequate setting which reduces $\theta_F$. Thus, we will investigate further in this regard towards a universal setting-up guide of $\alpha$ and $\nu$. Moreover, we will mathematically analyze the variation of selection/deletion probabilities in GA, aiming to boost evolutionary propagation during early generations.

REFERENCES

[1] J. H. Holland, “Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems,” Mach. Learn., vol. 2, pp. 593–623, 1986.

[2] R. J. Urbanowicz and J. H. Moore, “Learning classifier systems: A complete introduction, review, and roadmap,” J. Artif. Evol. Appl., vol. 2009, pp. 1–25, Sep. 2009.

[3] S.-J. Bu and S.-B. Cho, “A convolutional neural-based learning classifier system for detecting database intrusion via insider attack,” Inf. Sci., vol. 512, pp. 123–136, Feb. 2020.

[4] R. J. Preen, S. W. Wilson, and L. Bull, “Autoencoding with a classifier system,” IEEE Trans. Evol. Comput., vol. 25, no. 6, pp. 1079–1090, Dec. 2021.

[5] D. Howard, L. Bull, and P.-L. Lanzi, “A cognitive architecture based on a learning classifier system with spiking classifiers,” Neural Process. Lett., vol. 44, no. 1, pp. 125–147, Aug. 2016.

[6] R. J. Preen and L. Bull, “Deep learning with a classifier system: Initial results,” 2021, arXiv:2103.01118.

[7] M. Irfan, Z. Jiangbin, M. Iqbal, and M. H. Arif, “Enhancing learning classifier systems through convolutional autoencoder to classify underwater images,” Soft Comput., vol. 25, pp. 10423–10440, Mar. 2021.

[8] K. Matsumoto, R. Takano, T. Tatsumi, H. Sato, T. Kovacs, and K. Takadama, “XCS based on compressed input by deep neural network for high dimensional data,” in Proc. Genet. Evol. Comput. Conf. Companion, Jul. 2018, pp. 1418–1425.

[9] M. Irfan, Z. Jiangbin, M. Iqbal, Z. Masood, and M. H. Arif, “Knowledge extraction and retention based continual learning by using convolutional autoencoder-based learning classifier system,” Inf. Sci., vol. 591, pp. 287–305, Apr. 2022.

[10] R. Zhang, R. Stolzenberg-Solomon, S. M. Lynch, and R. J. Urbanowicz, “LCS-DIVE: An automated rule-based machine learning visualization pipeline for characterizing complex associations in classification,” 2021, arXiv:2104.12844.

[11] S. W. Wilson, “Classifier fitness based on accuracy,” Evol. Comput., vol. 3, no. 2, pp. 149–175, 1995.

[12] A. Stein and M. Nakata, “Learning classifier systems: From principles to modern systems,” in Proc. Genet. Evol. Comput. Conf. Companion, 2021, pp. 498–527.

[13] D. Pätzelt, A. Stein, and M. Nakata, “An overview of LCS research from IWLCS 2019 to 2020,” in Proc. Genet. Evol. Comput. Conf. Companion, Jul. 2020, pp. 1782–1788.

[14] K. Shafi and H. A. Abbass, “A survey of learning classifier systems in games,” IEEE Comput. Intell. Mag., vol. 12, no. 1, pp. 42–55, Feb. 2017.

[15] T. Kovacs, “Evolving optimal populations with XCS classifier systems,” School Comput. Sci., Univ. Birmingham, Birmingham, U.K., Tech. Rep. CSR-96-17 and CSR-96-17, 1996.

[16] T. Kovacs, “XCS classifier system reliably evolves accurate, complete, and minimal representations for Boolean functions,” in Soft Computing in Engineering Design and Manufacturing, Springer, 1998, pp. 59–68.

[17] T. Kovacs, “Towards a theory of strong overgeneral classifiers,” in Foundations of Genetic Algorithms. Amsterdam, The Netherlands: Elsevier, 2001, pp. 165–184.

[18] M. V. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson, “Toward a theory of generalization and learning in XCS,” IEEE Trans. Evol. Comput., vol. 8, no. 1, pp. 28–46, Feb. 2004.

[19] M. V. Butz, T. Kovacs, P. L. Lanzi, and S. W. Wilson, “How XCS evolves accurate classifiers,” in Proc. 3rd Genetic Evol. Comput. Conf., 2001, pp. 927–934.

[20] M. V. Butz and S. W. Wilson, “An algorithmic description of XCS,” Soft Comput. Fuzzion Found., Methodologies Appl., vol. 6, nos. 3–4, pp. 144–153, Jun. 2002.

[21] M. Nakata, W. Browne, and T. Hamagami, “Theoretical adaptation of multiple rule-generation in XCS,” in Proc. Genet. Evol. Comput. Conf., Jul. 2018, pp. 482–489.

[22] M. V. Butz and D. E. Goldberg, “Bounding the population size in XCS to ensure reproductive opportunities,” in Proc. Genet. Evol. Comput. Conf. Springer, 2003, pp. 1844–1856.

[23] M. Nakata, W. Browne, T. Hamagami, and K. Takadama, “Theoretical XCS parameter settings of learning accurate classifiers,” in Proc. Genet. Evol. Comput. Conf. Jul. 2017, pp. 473–480.

[24] M. Nakata and W. N. Browne, “Learning optimality theory for accuracy-based learning classifier systems,” IEEE Trans. Evol. Comput., vol. 25, no. 1, pp. 61–74, Feb. 2021.

[25] J. Hurst and L. Bull, “A self-adaptive XCS,” in Proc. Int. Workshop Learn. Classifier Syst. Springer, 2001, pp. 57–73.

[26] M. Horiiuchi and M. Nakata, “Self-adaptation of XCS learning parameters based on learning theory,” in Proc. Genet. Evol. Comput. Conf. Jun. 2020, pp. 342–349.

[27] M. Troč and O. Unold, “Self-adaptation of learning rate in XCS working in noisy and dynamic environments,” Comput. Hum. Behav., vol. 27, no. 5, pp. 1535–1544, Sep. 2011.

[28] R. Ferjani, L. Rejeb, and L. B. Said, “Belief extended classifier system: A new approach for dealing with uncertainty in sleep stages classification,” in Proc. Int. Conf. Hybrid Intell. Syst. Springer, 2020, pp. 454–463.
[29] T. Tatsumi, T. Komine, H. Sato, and K. Takadama, “Handling different level of unstable reward environment through an estimation of reward distribution in XCS,” in Proc. IEEE Congr. Evol. Comput. (CEC), May 2015, pp. 2973–2980.

[30] M. V. Butz, D. E. Goldberg, and K. Tharakunnel, “Analysis and improvement of fitness exploitation in XCS: Bounding models, tournament selection, and bilateral accuracy,” Evol. Comput., vol. 11, no. 3, pp. 239–277, Sep. 2003.

[31] M. V. Butz, D. E. Goldberg, and P. L. Lanzi, “Computational complexity of the XCS classifier system,” in Foundations of Learning Classifier Systems. Springer, 2005, pp. 91–125.

[32] A. Orriols-Puig and E. Bernadó-Mansilla, “Bounding XCS’s parameters for unbalanced datasets,” in Proc. 8th Annu. Conf. Genetic Evol. Comput. (GECCO), 2006, pp. 1561–1568.

[33] A. Orriols-Puig, E. Bernadó-Mansilla, D. E. Goldberg, K. Sastry, and P. L. Lanzi, “Facetwise analysis of XCS for problems with class imbalances,” IEEE Trans. Evol. Comput., vol. 13, no. 5, pp. 1093–1119, Oct. 2009.

[34] M. V. Butz, D. E. Goldberg, and P. L. Lanzi, “Effect of pure error-based fitness in XCS,” in Learning Classifier Systems. Springer, 2003, pp. 104–114.

[35] A. Orriols-Puig and E. Bernadó-Mansilla, “Revisiting UCS: Description, fitness sharing, and comparison with XCS,” in Learning Classifier Systems. Springer, 2006, pp. 96–116.

[36] R. Urbanowicz and J. Moore, “Retooling fitness for noisy problems in a supervised michigan-style learning classifier system,” in Proc. Ann. Conf. Genetic Evol. Comput., Jul. 2015, pp. 591–598.

[37] T. Kovacs and R. Tindale, “Analysis of the niche genetic algorithm in learning classifier systems,” in Proc. 15th Annu. Conf. Genetic Evol. Comput. Conf. (GECCO), 2013, pp. 1069–1076.

[38] F. Kharbat, L. Bull, and M. Odeh, “Revisiting genetic selection in the XCS learning classifier system,” in Proc. IEEE Congr. Evol. Comput., vol. 3, Sep. 2005, pp. 2061–2068.

[39] A. R. M. Wagner and A. Stein, “Adopting lexicase selection for michigan-style learning classifier systems with continuous-valued inputs,” in Proc. Genetic Evol. Comput. Conf. Companion, Jul. 2021, pp. 171–172.

[40] M. V. Butz, K. Sastry, and D. E. Goldberg, “Strong, stable, and reliable fitness pressure in XCS due to tournament selection,” Genetic Program. Evolvable Mach., vol. 6, no. 1, pp. 53–77, Mar. 2005.

[41] L. Bull, “A simple accuracy-based learning classifier system,” Learn. Classifier Syst. Group, Univ. West England, Bristol, U.K., Tech. Rep. UWELCSG03-005, 2003.

[42] L. Bull, “Two simple learning classifier systems,” in Foundations of Learning Classifier Systems. Springer, 2005, pp. 63–89.

[43] E. Bernadó-Mansilla and J. M. Garrell-Guix, “Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks,” Evol. Comput., vol. 11, no. 3, pp. 209–238, Sep. 2003.

[44] S. W. Wilson, “Get real! XCS with continuous-valued inputs,” in Proc. Int. Workshop Learn. Classifier Syst. Springer, 1999, pp. 209–219.

[45] S. W. Wilson, “Mining oblique data with XCS,” in Proc. Int. Workshop Learn. Classifier Syst. Springer, 2000, pp. 158–174.

[46] M. Iqbal, W. N. Browne, and M. Zhang, “Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems,” IEEE Trans. Evol. Comput., vol. 18, no. 4, pp. 465–480, Aug. 2014.

[47] E. Debie and K. Shafi, “Implications of the curse of dimensionality for supervised learning classifier systems: Theoretical and empirical analyses,” Pattern Anal. Appl., vol. 22, no. 2, pp. 519–536, May 2019.

[48] M. V. Butz and M. Pelikan, “Studying XCS/BOA learning in Boolean functions: Structure encoding and random Boolean functions,” in Proc. 8th Annu. Conf. Genetic Evol. Comput., 2006, pp. 1449–1456.

[49] I. M. Alvarez, W. N. Browne, and M. Zhang, “Reusing learned functionality in XCS: Code fragments with constructed functionality and constructed features,” in Proc. Companion Publication Annu. Conf. Genetic Evol. Comput., 2014, pp. 969–976.

[50] I. M. Alvarez, W. N. Browne, and M. Zhang, “Human-inspired scaling in learning classifier systems: Case study on the n-bit multiplexer problem set,” in Proc. Genetic Evol. Comput. Conf., 2016, pp. 429–436.

[51] M. V. Butz, M. Pelikan, X. Llorà, and D. E. Goldberg, “Automated global structure extraction for effective local building block processing in XCS,” Evol. Comput., vol. 14, no. 3, pp. 345–380, Sep. 2006.

[52] M. Abedini and M. Kirley, “An enhanced XCS rule discovery module using feature ranking,” Int. J. Mach. Learn. Cybern., vol. 4, no. 3, pp. 173–187, Jun. 2013.

[53] M. Horiiuchi and M. Nakata, “Extended learning optimality theory for the XCS classifier system on multiple reward scheme,” in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Dec. 2021, pp. 1–8.

RUI SUGAWARA received the B.E. degree from Yokohama National University, Japan, in 2019. He is currently a Graduate Student at the Graduate School of Engineering Science, Yokohama National University. His research interests include evolutionary rule-based learning and its theoretical analysis.

MASAYA NAKATA (Member, IEEE) received the Ph.D. degree in informatics from the University of Electro-Communications, Japan, in 2016. He is currently an Associate Professor at the Faculty of Engineering, Yokohama National University, Japan. He has been mainly working on evolutionary machine learning, data mining, and more specifically, theoretical analysis of evolutionary rule-based learning. Since 2019, he has focused his research on surrogate-assisted evolutionary algorithms. His contributions have been published through more than 15 journal articles and 35 conference papers, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, GECCO, and PPSN.