Classifier Generalization for Comprehensive Classifiers Subsumption in XCS

Caili Zhang
The University of Electro-communications
Chofu, Tokyo,Japan
zairitu@gmail.com

Takato Tatsumi
The University of Electro-communications
Chofu, Tokyo,Japan
tatsumi@uec.ac.jp

Hiyoyuki Sato
The University of Electro-communications
Chofu, Tokyo,Japan
h.sato@uec.ac.jp

Tim Kovacs
University of Bristol
United Kingdom
tim.kovacs@bristol.ac.uk

Keiki Takadama
The University of Electro-communications
Chofu, Tokyo,Japan
keiki@hc.uec.ac.jp

ABSTRACT

We proposed XCS-VRc^3 that can extract useful rules (classifiers) from data and verify its effectiveness. The difficulty of mining real world data is that not only the type of the input state but also the number of instances varies. Although conventional method XCS-VRc is able to extract classifiers, the generalization of classifiers was insufficient and lack of human readability. The proposed XCS-VRc^3 incorporating "generalization mechanism by comprehensive classifier subsumption" to solves this problem. Specifically, (1) All classifiers of the matching set subsume other classifiers, (2) Abolition of the inappropriate classifier deletion introduced by XCS-VRc (3) Preferentially select classifier with small variance of output in genetic algorithm. To verify the effectiveness of XCS-VRc^3, we applied on care plan planning problem in a nursing home (in this case, identifying daytime behavior contributing to increase the ratio of deep sleep time). Comparing the association rules obtained by Apriori, and classifiers obtained by XCS-VRc^3, the followings was found. First, abolishing the inappropriate classifier deletion and comprehensively subsuming promotes various degrees of generalization. Second, parent selection mechanism can obtain classifiers with small output variance. Finally, XCS-VRc^3 is able to extract a small number classifiers equivalent to large number of rules found in Apriori.

CCS CONCEPTS

• **Computing methodologies → Rule learning**;

KEYWORDS

LCS, XCS, Data mining

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07...$15.00
https://doi.org/10.1145/3205651.3208260

ACM Reference Format:
Caili Zhang, Takato Tatsumi, Hiyoyuki Sato, Tim Kovacs, and Keiki Takadama. 2018. Classifier Generalization for Comprehensive Classifiers Subsumption in XCS. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3205651.3208260

1 INTRODUCTION

In the context of data-mining research domain, its main purpose is to extract rules which derive appropriate outputs according to input data. In particular, it is important to find generalized rules that can cope with a lot of input data. Among many data-mining methods in the supervised learning, the methods for the classification problem (such as decision tree[8] or support vector machine[2]) work well when outputs can be classified as discrete classes, while those for the regression problem (such as linear regression[7] or neural network work[5]) well when outputs are continuous values. However, there is no method that automatically classifies the appropriate number of classes in outputs (Note that the unsupervised learning focuses on classifying inputs but does not focus on classifying outputs).

The mainstream framework of LCS is accuracy based LCS (XCS) [10], which can extract the most accurate and generalized rules. In related works, YCSc(a simple LCS for clustering)[11] was proposed for clustering in the XCS framework, but it does not have the concept of output clustering, and solves given problem by grouping (clustering) input data unlike our target which clusters output data.

XCS-VRc[12] can acquire rules whose range of acquired reward is less than a certain size, and it is expected to be applied to data mining of real-world problems,since the number of data differs between clusters and data is lack. To extract different rules with various generalitywith classifying appropriate numbers of classes at the same time, this paper proposes XCS-VRc^3 (XCS based on Variance of Reward for clustering a variance of reward based XCS for clustering and classifier compaction) which introduces the mechanisms of appropriately clustering outputs.

In this study, we focused on the data mining problem which leads the sleep condition from the behavior record of the elderly in the nursing home. The degree of sleep is evaluated at a ratio of deep sleep time. However, it is necessary to define classifiers to be acquired in order to evaluate the effectiveness of the method, in

Table 1: parameters of classifier

Symbol	*Pronounce*	*Meaning*
P	prediction	estimated payoff
ε	prediction error	error between prediction and payoff
F	fitness	the accuracy of prediction
num	numerosity	the number of the same classifier

which is unsupervised data. In this research, rules to be acquired are set by using the Apriori method which is a standard method in data mining. However, the number of rules that can be acquired by Apriori is enormous, it is necessary to judge what kind of classifier is significant from human perspective. In this research, we aim to surely acquire significant rules that Apriori acquired by XCS-VRc^3.

This paper is organized as follows. In Section 2, we introduce the brief mechanism of XCS. We introduce the framework of XCS-VRc in Section 3. The proposed XCS-VRc^3 is introduced in section 4, Section 5 explains the data ming problem we tackled. In Section 6 explains the experiment setting and results. Section 7 we make discussion about the result of experiment. And Section 8 provides our conclusions and present future work.

2 ACCURACY-BASED LCS

The XCS classifier system maintains a population of classifiers which represent the solution to a reinforcement learning problem [6]. The following sub-Sections explain the classifiers and mechanisms of XCS.

2.1 Classifier

A classifier is a *condition-action* rule; a condition C is coded by $C \in \{0, 1, \#\}^L$, where L is the length of condition, and the symbol '#' is the *don't care symbol* which matches all input values (*i.e.*, 0 or 1). Classifiers consist of a condition, an action, and four main parameters as followed [10] [3] as shown in Table 1:

2.2 Mechanism

At iteration t, XCS builds a *match set* [M] containing the classifiers in the population [P] whose condition matches the current sensory input s_t; if [M] does not contain all the feasible actions *covering* takes place and creates a set of classifiers that matches s_t and cover all the missing actions. This process ensures that XCS can evolve a complete mapping so that in any state it can predict the effect of every possible action in terms of expected returns.[1]

For each possible action a in [M], XCS computes the *system prediction* $P(s_t, a)$ which estimates the payoff that XCS expects if action a is performed in s_t. The system prediction $P(s_t, a)$ is computed as the fitness weighted average of the predictions of classifiers in [M] which advocate action a:

$$P(s_t, a) = \sum_{cl_k \in [M](a)} p_k \times \frac{F_k}{\sum_{cl_i \in [M](a)} F_i} \tag{1}$$

where, $[M](a)$ represents the subset of classifiers of [M] with action a, p_k identifies the prediction of classifier cl_k, and F_k identifies the fitness of classifier cl_k. Then XCS selects an action to perform; the classifiers in [M] which advocate the selected action form the current *action set* [A].

The selected action a_t is performed, and a scalar output r_t is returned to XCS

When the output r_t is received, the estimated payoff $P(t)$ is computed as follows:

$$P(t) = r_t + \gamma \max_{a \in [M]} P(a) \tag{2}$$

where γ is the discount factor [9]. Next, the parameters of the classifiers in [A] are updated in the following order [3]: prediction, prediction error, and finally fitness. Prediction p is updated with learning rate β ($0 \leq \beta \leq 1$):

$$p_k \leftarrow p_k + \beta(P(t) - p_k) \tag{3}$$

The absolute accuracy κ is defined by equation (5). Then, the prediction error ϵ and classifier fitness are updated with relative accuracy κ':

$$\epsilon_k \leftarrow \epsilon_k + \beta(|P(t) - p_k| - \epsilon_k) \tag{4}$$

$$\kappa(cl) = \begin{cases} 1 & \text{if } \epsilon < \epsilon_0 \\ \alpha \left(\frac{\epsilon}{\epsilon_0}\right)^{-\nu} & \text{otherwise} \end{cases} \tag{5}$$

$$F_k \leftarrow F_k + \beta(\kappa'_k - F_k) \tag{6}$$

On a regular basis (dependent on parameter θ_{GA}), the genetic algorithm is applied to classifiers in [A]. It selects two classifiers, copies them, and with probability χ performs crossover on the copies; then, with probability μ it mutates each allele. XCS often employs tournament selection with the tournament size τ [4] as a selection mechanism that improves the performance of XCS [4].

3 XCS-VRC

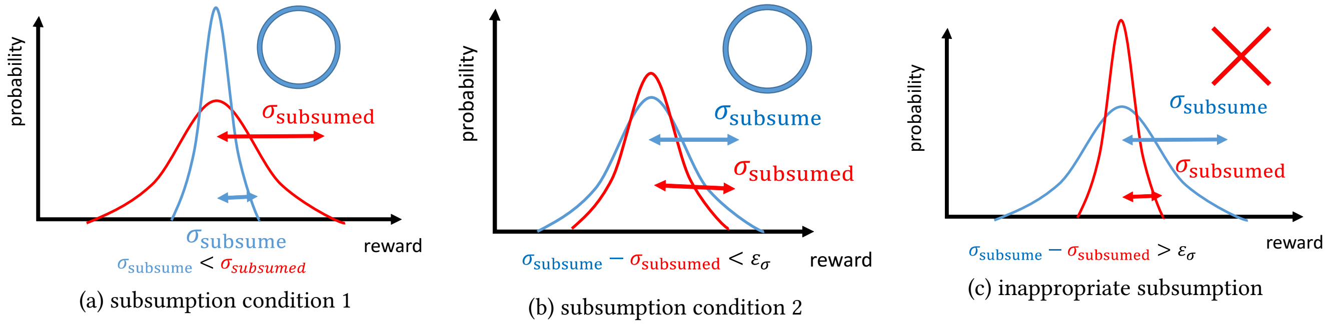
XCS-VRc has the following three mechanisms to generalize rules according to the different number of classes. (1) the estimation mechanism of the standard deviation of the acquired output to generate and keep the rules. (2) the detection mechanism for the over generalized rules which has a too large error range. (3) the new subsumption mechanism which prohibits to absorb the rules in a certain level class by the (generalized) rules in higher level class.

3.1 Estimation mechanism

Firstly calculating ϵ_0 state-action (Rule)'s deviation must converge. Because a unstable value of state-action (Rule)'s deviation will lead to inaccurate result of estimation of ϵ_0. Secondly, XCS-VRc recalculates classifier's ϵ_0 in equation (7) to cope with difference of average of output to obtain generalized rules. Since XCS-VR does not consider the different between multiple state-action (Rule)'s average output in estimating the classifiers accuracy criterion ϵ_0, XCS-VR regards generalized rule as bad classifier then delete it in the learning process.

Thus the accuracy of both specialized rules and generalized rules would be 1 according to equation (5) and could be remained during

[1] In the algorithmic description [3], covering is activated when the match set contains less than θ_{mna} actions; however, θ_{mna} is always set to the number of available actions so that the match covers all the actions.



(a) subsumption condition 1
(b) subsumption condition 2
(c) inappropriate subsumption

Figure 1: subsumption condition

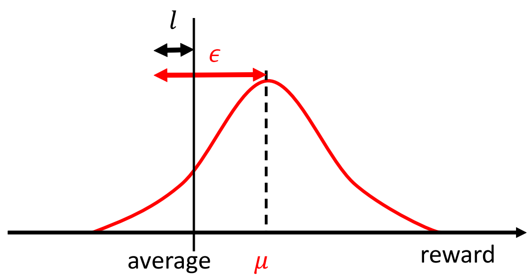


Figure 2: Extraction policy

the learning process.

$$cl.\varepsilon_0 \leftarrow \sqrt{\frac{(\sum_{MatchedRules} n_i S_i^2 + n_i(\bar{\mu} - \mu_i)^2)}{(\sum_{MatchedRules} n_i)}} \tag{7}$$

3.2 Detection mechanism

The XCS-VRc intent to extract both generalized and specialized rules. Considering interpret ability in real world application, not every generalized classifier is useful. Obviously classifiers like "####" which error might cover whole range of output should be deleted. However, due to applying equation (7), such over generalized rules remained in the results. In order to delete over generalized rules, a deletion criterion must be defined. Thus in this paper, we determine that classifiers which prediction error's covered range crossed average line of whole output should be deleted. As shown in Fig.2, Length l is classifier's ϵ range cross over average, the longer l is, the lower classifier's accuracy will be evaluated. On this situation classifier's accuracy κ is updated by equation(8), other wise κ is updated by equation (5) as same as conventional methods.

$$\kappa_j \leftarrow \alpha(1 + \frac{l}{\varepsilon_j})^{-\nu} \tag{8}$$

3.3 New subsumption mechanism

Conventional subsumption mechanism is designed to promote integration of classifiers. In XCS and XCS-VR the subsumption is applied in *action set* [A]. when there is a classifier more generalized than others, this classifier can subsume other classifiers by deleting other classifiers from [P] and add other classifiers' numerosity n to itself. More specifically, cl_a = "#0#" is more generalized than cl_b = "10#", since cl_a has more "#" than cl_b, besides "#0#" also matches cl_b's "10#", thus cl_a can subsume cl_b, which means delete cl_b then add cl_b's numerosity to cl_a.

However, in this paper, the purpose is acquiring both specialized (e.g "10#") and generalized rules (e.g. "#0#") in parallel.

As shown in Fig. 1(c) specialized classifier with a narrower error range ϵ is not allowed to be subsumed by generalized rule with a broader one. Thus, the criterion θ_σ must be defined to distinguish specialized rules from generalized ones. The XCS-VRc determined $\theta_\sigma = 1/10 \times \epsilon_0(subsume)$ that distinguishes specialized classifiers from generalized classifiers as shown equation (11).

As shown in Fig. 1(a) and Fig. 1(b), if the difference between classifier(subsume)'s error range and classifier(subsumed)'s error range within in the criterion as shown in equation (10), we consider the 2 classifiers are similar with each other, and can be combined.

If a classifier is more generalized than other classifiers and satisfied either of the 2 situations, it can subsume other classifiers. While equation (9) can be combined with equation (10).

$$cl.\epsilon_0(subsume) < cl.\epsilon_0(subsumed) \tag{9}$$

$$cl.\epsilon_0(subsume) < cl.\epsilon_0(subsumed) + \theta_\sigma \tag{10}$$

$$cl.\epsilon_0(subsume) > cl.\epsilon_0(subsumed + \theta_\sigma \tag{11}$$

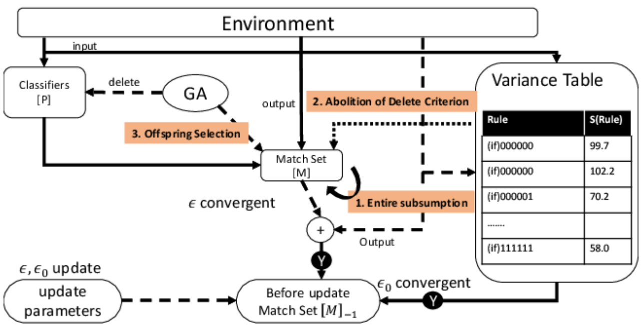
3.4 Algorithm

The main flow of XCS-VRc is mostly the same as XCS-VR. XCS-VRc is processed as follow. Firstly, XCS-VRc revives input state from environment, then selects classifier matched the input state from population [P] to form match set [M]. In [M] XCS-VRc calculate the prediction output of each action and store the results in prediction array PA, then select classifier with the specific action to forms the action in proportion to the value of PA. Secondly, the variance table is updated by the mechanism described. Finally, In action set [A], once the output of classifier is converged, the standard deviation ϵ_0 of the outputs in classifiers is updated by estimation mechanism described in section 4.1. The accuracy of each classifier is updated by the detection mechanism described in section 3.2. The subsumption is applied on action set by the new subsumption mechanism described in section 3.3 once the subsuming classifier meets the subsumption condition.

4 XCS-VRc^3

4.1 Architecture

XCS-VRc^3(XCS based on Variance of output for Clustering and Classifier Compaction) inherits the framework of XCS-VRc, The changes of XCS-VRc^3 and XCS-VRc of the conventional method are as follows. (1) By integrating subsumption on [M] for all classifiers, classifiers of various degrees of generalization are integrated. (2) Excludes criteria for deletion of classifiers determined by the average of overall output. (3) By making it easy for the ϵ small classifier to be selected as a parent, we obtain a classifier with a low generalization degree. The mechanism of XCS-VRc^3 will be described in detail below.



Rule	S(Rule)
(if)000000	99.7
(if)000000	102.2
(if)000001	70.2
....	
(if)111111	58.0

Figure 3: XCS-VRc3 Architecture

4.2 Memory component

Classifiers and Population

XCS-VRc^3, like XCS-VRc, classifiers include predicate output value p, goodness of fit F degree of polymerization n etc. It has elements of. A mechanism that preserves the standard deviation of the output of *Rule* that incorporates all the environmental conditions and the actions of the behavioral unit that the condition part matches. ϵ_0 obtained from the Variance Table, the sample standard of the output value acquired by the classifier Save deviation s

4.3 Mechanism

4.3.1 Entire Classifiers Subsumption Mechanism. We were able to acquire both classifiers with a high degree of generalization and classifiers with a low degree of generalization in the subsumption mechanism of XCS-VRc. The problem is that classifiers with different generalizations are not fully integrated, the number of classifiers becomes huge.

Conventionally XCS-VRc's inclusion integrates classifiers that can be included by extracting the most generalized classifier in [M] XCS-VRc^3 encloses from the most generalized classifier to [M], then subsumes all classifiers in descending order of generalization degree . Figure ref fig: xcs-vrc3-mecha1-image represents the whole subsumption image, the list contains classifiers in the collation group in descending order of generalization, left side conventional XCS-The subsumption mechanism of VRc subsumes all other classifiers that satisfy the subsumption condition, with Subsumer being the most generalized first classifier. However, classifiers 3 and 4 are not included in number 1, but number 4 can be included in classifier 3, but it is not sufficiently subsumed. on the other hand XCS-VRc^3 registers all classifiers in the Subsumer sequentially from the order of high generalization degree, and subsumes. As a result, the third classifier can subsume 4.

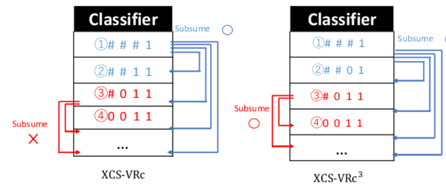


Figure 4: Image of Entire Classifiers Subsumption Mechanism

4.3.2 Abolition of delete criterion. Conventional XCS-VRc can eliminate over-generalized classifiers by extremely lowering the accuracy of classifiers across averages, but adverse effects that can not be acquired if classifiers that you want to obtain are near the overall average are is there. However, since this setting is arbitrary, XCS-VRc^3 excluded this.

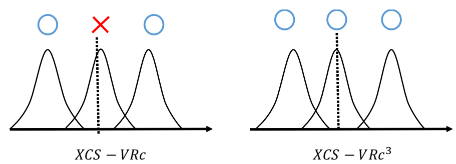


Figure 5: Image of Mechanism 2

Also exclude classifiers where all attributes are #, because human beings have no significance because classifiers with all attributes # have no features. Also, when subsuming, classifiers with all attributes # may misclassify other classifiers. Cover, GA crossover and mutation Detect and eliminate classifiers with all attributes # from the mechanism that generates these three classifiers

4.3.3 Variance Based Offspring Selection Mechanism. We would like to acquire classifiers with small variance and good prediction accuracy with few inputs corresponding to classifiers with low generalization degree explained in XCS-VRc. However, when XCS-VRc is applied to real data, classifiers with a low generalization degree can not be acquired. Because [M] has a small number of classifiers that are not generalized, it can not be selected by parent selection, and as a result it is deleted from [P]. Conventional XCS-VRc is easy to choose classifier with high degree of conformity as a parent, based on goodness of fit. The evaluation of concrete parent selection is calculated based on the expression 12.

$$Classifier_{offspring} = \left(\frac{1}{Classifier\epsilon_0}\right)^5 \tag{12}$$

Since XCS-VRc^3 wants to acquire classifiers with small variance, we changed the criteria for choosing parents to the classifier 's prediction error range ϵ_0.

4.4 Algorithm

Algorithm 1 shows the algorithm of XCS-VRc^3, in one step of learning it follows the following procedure. For lines 3 to 5, the problem assumed by XCS-VRc^3 is no action like XCS-VRc, In the 12 to 26 lines, the output value is given to the classifier from the data set, and in the reinforcement learning section, the evaluation of the prediction output and classifier is updated with the output acquired. The changes of XCS-VRc^3 and XCS-VRc are Line 8 subsumes all classifiers of [M] by the mechanism of 4.3.1 Line 9 selects the parent by the mechanism of the 4.3.2 clause. In row 22, the classifier is generated and culled by classifier evaluation in the discovery section GA.

Algorithm 1 XCS-VRc^3 Algorithm

```
1:  while (! end of iterations) do
2:    while (! end of problem) do
3:      state ⟵ environment :
4:      generate [M]: [P]
5:      output ⟵ environment:
6:      if [M]−1 ≠ null then
7:        parameter update: [M]−1
8:        [M].Subsumbe() : [M] Subsumption
9:        Parent = [M].SelectOffspring() : [M] Select Offspring based on ϵ
10:       run GA : [M]−1
11:     end if
12:     if end of problem then
13:       parameter update: [M]
14:       for all classifier cl in [M] do
15:         if convergence of S(Rule) then
16:           cl.ϵ0 update: Rule
17:         end if
18:         if convergence of cl.s then
19:           cl.ϵ ⟵ cl.s:
20:         end if
21:       end for
22:       run GA: [M]
23:     end if
24:     set [M]−1 : [M]−1ãĄń[M]
25:   end while
26: end while
```

5 REAL DATA MINING PROBLEM

In this paper we tackle a real world data mining problem: extracting rules lead to deep/light sleep from a dataset consist of the activity records and sleep data. The reason we focus on this problem is (i)It is difficult for many elderly people to have a deep sleep(The index of deep sleep is the ratio of deep sleep (REM sleep 3 and 4) to sleep time), while deep sleep is necessary to their health, (ii)It is essential for enhancing the quality of life to generate a better care plan, because activities such as, exercise or rehabilitation contribute to the depth of sleep. We proposed XCSI-VRc^3 to tackle this problem. To verify the effectiveness of XCSI-VRc^3, care plan(activities) data and sleep depth records of diabetic women at age 82 at nursing care facilities (September 2011 to 2012 March) are used for learning. The elements of care plan data and their meanings are shown in Table 2, and one example of input data is shown in Table 3.

5.1 Association Rules and Apriori

5.1.1 Association Rules. Association rules mining is aim at discovering interesting relations between variables in instances of large databases. The rule of combination of frequently occurring items in these instance is called association rule. In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best-known constraints are minimum thresholds on support and confidence and lift.

Support is the ratio of the number of instances that satisfy X and Y of the total instances number and is calculated by the equation (13). M is the total number of instances, and $\sigma(x)$ is the number of instances that satisfy x.

$$\text{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|} \tag{13}$$

Confidence is the ratio of the number of transactions that satisfy X and Y among instances that satisfy X, and is calculated by the expression (14).

$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X) \tag{14}$$

5.1.2 Apriori Algorithm. It is difficult to calculate all association rules when a large amount of data is targeted. Apriori algorithm [1] is a method for narrowing down the search space and accelerating search speed using support and confidence constrains. The Apriori algorithm searches for association rules in the following two steps.

(1) Since a instance that satisfies both conditions X and Y always satisfies the condition X, $supp(X \cup Y) \leq supp(X)$ holds. Therefore, a set ($supp(X) \geq minsupp$) that satisfies the minimum support ($minsupp$) is extracted as set F of frequent items.
(2) In $X \Rightarrow Y$, extracting rules that satisfy the conditions $X \cup Y \in F$ and $conf(X \Rightarrow Y) \geq minconf$, where $minconf$ is the minimum confidence threshold. Thus association rules are extracted.

5.2 Required Rules

Since we want to reveal the pattern of activities lead to high/low sleep depth, association rule mining method aproach is considerable. A priori which is a general method of data mining can efficiently extract valid rules from all combinations of rules. However, valid rules are determined by the support and reliability set by humans,

With a support of 0.1 and a confidence greater than 0.6. In other words, all the rules leading to bad sleep were extracted with the probability that the occurrence frequency was more than 10% and the rule is 60% correct for the data set. Since this result has a large number of rules, it is necessary to summarize these rules to simpler rules.

Since the most distinctive difference of rules leading to good and bad sleep is the row of bathing and rehabilitation, we focus on these two columns. Rules leading to good sleep are rules which has bathing attribute= 2(pm) and rehabilitation=blank (optional), or bathing=blank and rehabilitation= 0(not do). On the other hand,

Table 2: Meaning of elements of input data

	Wakeup	sleeping	tea	gardening	bathing	snacks	newspapers	rehabilitation
0	early	early	none	none	none	none	none	none
1	normal	normal	am	am	am	am	am	am
2	late	late	pm	pm	pm	pm	pm	pm
3	not sleeping	not sleeping	both	both	both	both	both	both

Table 3: An example of input data

	Wakeup	sleeping	tea	gardening	bathing	snacks	newspapers	rehabilitation	sleep depth
input	1	0	0	0	2	0	0	1	18%
Meaning	ordinary	early	none	none	pm	none	none	am	

Table 4: Rules to Acquire Decided by Apriori

Classifier that leads to good sleep							
Wakeup time	bedtime	tea	gardening	bathing	side dishes	newspaper	rehabilitation
#	#	#	#	2	#	#	#
#	#	#	#	#	#	#	0
Classifier that leads to bad sleep							
Wakeup time	bedtime	tea	gardening	bathing	side dishes	newspaper	rehabilitation
#	#	#	#	0	#	#	1

rules leading to bad sleep are summarized in one rule which has bathing=0(do not take) and rehabilitation=1(am). The other columns are both included in good and bad sleep, can not be an element that distinguishes between good and bad sleep. In other words, our goal is to acquire the following three rules as classifiers.

6 EXPERIMENT

6.1 Experiment setting

Besides the output environment, parameters are the same as XCS-VR, and set as follow: $N = 400$, $\gamma = 0.71$, $\beta = 0.2$, $\epsilon = 10$, $\mu = 0.04$, $P_{\#} = 0.33$, $\chi = 0.8$, $\nu = 5$, $\theta_{GA} = 25$, $\theta = 20$, $\theta = 20$. While, XCS-VRc^3 share the same parameters as θ_{std}=15, θ_{ref} = 75, $E_0 = \epsilon_0 =$ 10. The learning iteration is 100,000 as 1 trial.

6.2 Experiment Result

6.2.1 Acquired Classifiers. Extracting the classifier acquired by XCS-VRc^3 with the criteria of convergence and accuracy=1, and show it in 4.Three kinds of classifiers we want to acquire were confirmed. Classifiers that leads to good sleep The average sleep depth and variance value of the two classifiers(yellow) are relatively high, and the average sleep depth and variance of the classifier(blue) leading to poor sleep is relatively low.

6.2.2 Comparison. Table 8 also shows specialized classifiers acquired by XCS- $rmVRc^3$.Although the number of input data is small, it is possible to abtain care plan with only one instance as classifier, and mark the unimportant attributes as wild card #. However, this classifier is a classifier limited to this data set.

XCS-VRc^3 shows classifiers leading to good/bad sleep in table **??**. Looking at the classifier that leads to good/bad sleep (the predicted output value is higher/lower than the average value 0.143), focusing on bathing and rehabilitation, baths in the pm and rehabilitation is optional when deep sleep It is understood that it leads. On the other hand, we had rehabilitation in the morning and found a relationship leading to shallow sleep when not taking a bath.

7 ANALYSIS

7.1 Entire Classifiers Subsumption Mechanism

only applied the first mechanism, and analysis the results. problem: Classifiers whose bathing is 0 rehabilitation is 1 are not compiled. There are several classifiers with the same mean and variance in the result of XCS-VRc, there are still subsumable classifiers, because subsumption is done for the most generalized classifier, Many of the classifiers are not subsumed and remain intact.

On the other hand, XCS-VRc^3 subsumes not only the most generalized classifier but also the whole of [M] in order of generalization degree.

Classifiers with bathing 0 rehabilitation 1 were gathered and earned.

However, it became impossible to acquire a classifier that leads to sleep, which is defined as the table 4 and the bathing to be acquired is # and the rehabilitation is good.

7.2 Disuse of Average output Delete Criteria

In order to exclude overly generalized classifiers, the conventional XCS-VRc initially did not acquire classifiers across the average of the entire subsumption. Classifier bathing # rehabilitation is no longer able to acquire classifier leading to good sleep, because the average and variance of the input data matched by this classifier cross over the average value of the overall output 0.143, so in the

Table 5: Acquired Classifiers

wake	sleep	tea	garden	bath	snack	news	Rehabilitation	mean	epsilon
#	#	#	#	**2**	#	#	#	**0.150**	**0.024**
#	#	#	#	#	#	#	**0**	**0.147**	**0.028**
#	#	#	#	**0**	#	#	**1**	**0.134**	**0.024**

Table 6: delete reason

Classifier	iteration	sleep depth	variance	fitness	numerosity	[M]	accuracy	generate time	delete reason
####0##1	28400	0.134	0.024	0.003	1	668	1	25920	delete in [P]
####0##1	84408	0.134	0.024	0.003	1	525	1	82451	delete in [P]
####0##1	93960	0.000	0.027	0.000	1	0	1	93960	GA subsumption
####0###	93960	0.140	0.027	0.060	21	54490	1	2557	GA subsumption

Table 7: bath=0 rehabilitation=1 offspring selection in[M]

classifier	fitness	selection possibility
####0##1	0.099	0.100
######2#	0.126	0.128
####0###	0.114	0.115
#######1	0.105	0.107
#0######	0.139	0.141
1#######	0.120	0.121
#####0##	0.115	0.116
###2####	0.100	0.101
##2#####	0.061	0.062
##2###2#	0.008	0.008

ϵ_0	selection possibility
0.025	0.192
0.028	0.110
0.028	0.104
0.028	0.100
0.029	0.097
0.029	0.095
0.029	0.093
0.029	0.088
0.031	0.060
0.031	0.060

Table 8: Specialized classifier

Specialized classifier											
Wake up time	bedtime	tea	gardening	bathing	side dishes	newspaper	rehabilitation	average sleep depth	dispersion	fitness	generation time
#	#	#	1	#	#	#	#	0.147	0	0.033	30841
#	#	2	#	#	2	#	#	0.12	0	0.009	16298

learning process Although it was generated, it was evaluated low in accuracy and deleted in [P].

XCS-VRc^3 removes the over generalized classifier of XCS-VRc to exclude classifiers that span overall output, Excluding the mechanism to do learning.

The classifier obtained by XCS-VRc^3 is shown in ref table: no - heikin. XCS-VRc^3 was able to acquire the classifier whose bathing # rehabilitation was 1. However, it is impossible to acquire the classifier that leads to a bad sleep with 0 rehabilitation being 1 bathing this time. As shown in the reason table 6, the classifier "####0##1" you want to acquire is generated but it is chosen as the parent There is no opportunity, the degree of polymerization can not be increased by 1, because the degree of relevance is low [P] It is deleted, and when it was generated it may have been included in parents immediately.

7.3 Variance Based Select Offspring

Bathing 0 Rehabilitation is 1 When choosing parents with GA, you can not select a classifier, the degree of polymerization has been one and the fitness is low. Classifiers with low relevance are often deleted with [P], so they are repeatedly generated and disappeared.

XCS-VRc^3 changed the criteria for choosing parents, the bathing is 0 rehabilitation is 1 The correspondence of the input corresponding to the classifier is relatively low The classifier of this class is outputted Because I predict it relatively accurately, I want to acquire from other classifiers. Table 7 shows an example of a matching group including bathing 0 rehabilitation classifier 1, bathing 0 classification of 0 rehabilitation is 1 best fitness value Since it is not large, it is deleted by learning assumption because there is little probability that it can be selected as a parent in the parent selection criteria of the conventional XCS-VRc. On the other hand, the base selection criterion of XCS-VRc^3 is calculated based on the expression 12, and since the evaluation of classifiers with small variance is high, the bathing is 0 rehabilitation is 1 Was the highest and the probability that it was chosen as the classifier parent was high. In the learning process, it was not deleted and it was acquired.

When choosing XCS-VRc^3 parent, the classifier with lower variance is set to dominate, and the probability of being chosen as a parent has risen. As a result, bathing in the learning process increased the degree of polymerization of classifiers with 1 rehabilitation, and as the fitness increased, it was not deleted by [P]. The result of XCS-VRc^3 was that you could get a classifier (blue) with bathing 0 and rehabilitation 1.

8 CONCLUSION

In this research, we propose a learning classifier system XCS-VRc^3, which has few learning data and can cope with biased data sets for application to actual data. We introduced a new generalization mechanism to XCS-VRc of existing method. XCS-VRc^3 proposed the following three mechanisms. 1) the abolition of the mechanism to remove classifiers that take near the average value of the total output in XCS-VRc, 2) regardless of the generalization degree of classifiers, all classifiers included in the matching group are included in other classifier subsumption 3) By introducing a mechanism to make it easy for GA parents to select classifiers with a low degree of generalization, the distribution of evaluations will acquire various classifiers, In order to verify the effectiveness of XCS-VRc^3, when applied to elderly care plan and sleep data in nursing home, we got the following findings. 1, XCS-VRc^3 can stably evaluate classifiers even in an environment where the evaluation is scattered and be able to acquire all classifiers to be acquired, which are determined from the association rules obtained by Apriori. 2, XCS-VRc^3 properly generalized the classifier and it was shown that it is possible to extract rules from the data set with fewer classifiers than the association rules found by Apriori. 3. It was found that both generalized classifiers and specialized classifiers can be obtained for a small number of data

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 487–499. http://dl.acm.org/citation.cfm?id=645920.672836

[2] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92)*. ACM, New York, NY, USA, 144–152. https://doi.org/10.1145/130385.130401

[3] Martin V Butz and Joachim Hoffmann. 2002. Anticipations control behavior: Animal behavior in an anticipatory learning classifier system. *Adaptive Behavior* 10, 2 (2002), 75–96.

[4] Martin V Butz, Kumara Sastry, and David E Goldberg. 2003. Tournament selection: stable fitness pressure in XCS. (2003), 1857–1869.

[5] Trevor Hastie. 1998. *Neural network.* Wiley Online Library.

[6] Pier Luca Lanzi. 2002. Learning classifier systems from a reinforcement learning perspective. *Soft Computing* 6, 3-4 (2002), 162–170.

[7] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. 2015. *Introduction to linear regression analysis.* John Wiley & Sons.

[8] J. R. Quinlan. 1986. Induction of Decision Trees. *MACH. LEARN* 1 (1986), 81–106.

[9] Richard S Sutton. 1998. Between MDPs and Semi-MDPs: Learning, planning, and representing knowledge at multiple temporal scales. (1998).

[10] S.W.Wilson. 1995. Classier Fitness Based on Accuracy. *Evol. Comput* (1995), 149–175.

[11] Kreangsak Tamee, Larry Bull, and Ouen Pinngern. 2007. YCSc: A Modified Clustering Technique Based On LCS. *Journal of Digital Information Management* 5, 3 (2007).

[12] Caili Zhang, Takato Tatsumi, Masaya Nakata, and Keiki Takadama. 2017. Approach to Clustering with Variance-Based XCS (Special Issue on Cutting Edge of Reinforcement Learning and its Applications). *Journal of advanced computational intelligence and intelligent informatics* 21, 5 (sep 2017), 885–894. https://ci.nii.ac.jp/naid/40021316679/