Stay Real! XCS with Rule Combining for Real Values

Nugroho Fredivianus
Karlsruhe Institute of Technology
Institute AIFB
76128 Karlsruhe, Germany
nugroho.fredivianus@kit.edu

Kais Kara
Karlsruhe Institute of Technology
Institute AIFB
76128 Karlsruhe, Germany
karakais@hotmail.com

Hartmut Schmeck
Karlsruhe Institute of Technology
Institute AIFB
76128 Karlsruhe, Germany
hartmut.schmeck@kit.edu

ABSTRACT

This paper provides an extension of the rule combining (RC) technique in the Accuracy-based Learning Classifier System (XCS) to handle continuous-valued input. Previously implemented to cope with binaries, the suitability of the newly introduced algorithm is investigated in further tasks. Several experiments are run and the results are compared to previous work using the real-valued multiplexer problem. The comparison shows that by implementing the RC technique, real value XCS is capable of producing a compact population of rules through proper generalizations. Moreover, the learning rate of Real-value XCS-RC (RXCS-RC) is comparable or even superior, in some cases.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

Keywords

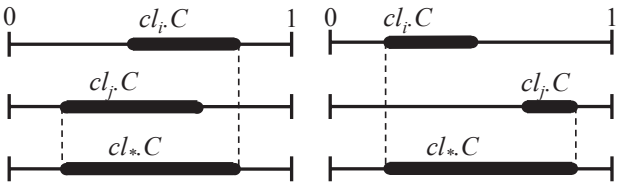
Learning, classifier systems, generalization

1. INTRODUCTION

The accuracy-based learning classifier system (XCS, [2]) is one of the most widely known online learning system in the field of Machine Learning. XCS stores its knowledge by maintaining some rules called classifiers in the population $[P]$. Each classifier cl has the attributes named condition $cl.C$, action $cl.A$, prediction reward $cl.P$, prediction error $cl.\varepsilon$ and some other housekeeping values.

A variant of XCS called XCS-RC, which uses the Rule Combining (RC) technique, has successfully solved some tasks using binary inputs for both single-step and multi-step mode [1]. Here, the investigated system (denoted as Real-value XCS-RC or RXCS-RC afterwards) is learning to generalize classifiers by creating new rules based on conclusions taken from the already existing rules. This is done without any involvements of random variables. The goal is to get a compact population filled with general rules (but not over-general) with high accuracy, while at the same time, fulfilling the expected system performance.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.



(a) Overlapping (b) Not overlapping

Figure 1: Combining a pair of intervals

2. THE RXCS-RC

In [3], Wilson introduced XCSR, the first variant of XCS coping with continuous-valued inputs. The learning system attempts to create accurate rules by implementing genetic operators, which is replaced by RC in RXCS-RC. As a result, seven parameters in XCSR are replaced by only four in RXCS-RC. Those four are the combining period (T_{comb}), minimum experience to be combined ($minExp$), prediction tolerance ($predTol$) and prediction error tolerance ($predErrTol$).

The learning cycle starts when an input x is received and the system seeks whether there are matching classifiers in $[P]$. In case no rules found, new ones are created in a so-called covering process. XCSR covers the value of x by setting it as the center of an interval within $cl.P$ and a random number $m \leq 0.5$ becomes the spread. Here, generalization makes the interval able to cover a wider range than the input itself. Different to that, RXCS-RC covers the inputs using dots, without generalization.

Then, the matching classifiers enter the match set $[M]$, including those been created via covering. After a process of action selection, the classifiers with the "winning" action are collected into the action set $[A]$ and their experience are incremented by one. Finally, a reward is received from the environment and is used to update the classifiers in $[A]$ before storing them back to the population.

Periodically after T_{comb} updates, RXCS-RC checks the classifiers in $[P]$ and picks some pairs of rules. To become a parent, a classifier must fulfill the $minExp$ constraint. Combining is only allowed if it does not violate the basic principles, which consist of three criteria:

$C1$. The parents advocate the same action

$C2$. The prediction rewards of the parents are not significantly different from each other

$C3$. There are no disproving rules in the population

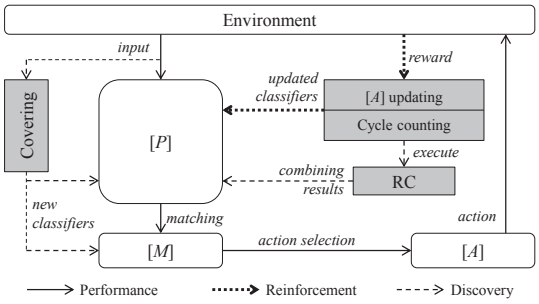


Figure 2: Learning cycle in XCS-RC

A child candidate cl_* is composed from a pair of rules fulfilling C1, while $predTol$ is used to check whether C2 is satisfied or not. Following criterion C3, the system operates an examination for any existence of disproving classifier. A rule is called a disproval if it is able to match the same input x and having the same output as cl_*, but the difference of their prediction is greater than $predTol$. If at least one disproval exists, cl_* will be discarded. Otherwise, it is accepted to enter $[P]$.

Figure 1(a) and 1(b) depict a pair of parents cl_i and cl_j having overlapping and not overlapping intervals, respectively. The combining result is given to the child (cl_*), where the smallest point is set as the lower bound and the biggest one becomes the upper constraint. The idea of this "heuristic-based evolution" is to infer knowledge from the parents by drawing conclusions based on the already existing classifiers.

RXCS-RC implements all concepts as the binary-mode XCS-RC (see Figure 2 and [1]). They mainly modify the discovery component of XCS, with an addition of a new deletion mechanism. This removal takes place after updating the classifiers in $[A]$, suffered by classifiers having intolerably high $cl.\varepsilon$ with the value of $predErrTol$ as the threshold.

Changes also occur on the subsumption-deletion method. XCSR subsumes rules in two occasions: when inserting new classifiers into $[P]$, and building $[A]$. RXCS-RC adopts this technique, but executes it after cl_* passed the disproval examination. Finally, as XCSR, RXCS-RC removes some rules when the size exceeds the limitation. However, experiments show that RC is capable of keeping the size always below the cap, using the parameters given to XCSR.

3. EXPERIMENTS AND RESULTS

The task of real-valued multiplexer is used to evaluate the learning effectiveness of RXCS-RC and the result is compared to that published in [3]. Both of the simulations use the same data acquisition method and parameter values. The correctness rate is compared to judge the learning progress, while the population size denotes the generalization capabilities. The given figure shows an average of 20 RXCS-RC runs. Due to space limitations, only the result of one multiplexer task is displayed here.

Figure 3 depicts the comparison between XCSR and RXCS-RC. Here, XCSR are superior in the first 2000 explore trials but then between 4000 to 7000 RXCS-RC indicates a better learning rate. Then finally, until the end of the simulation, both of them are comparable with around 99% in fulfilling the task correctly.

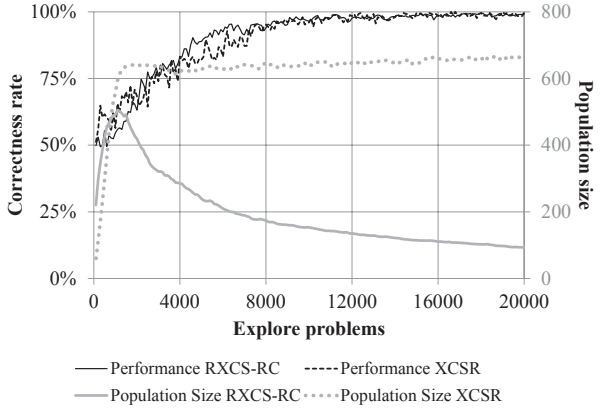


Figure 3: Comparisons on the system performance and the population size

Regarding the population size, RXCS-RC minimizes the number of rules below 160 after 10 000 explore problems. The usage never reaches 600 classifiers and is kept below it, which means that no rules were removed due to space limitation. It is significantly superior to XCSR, where the number of classifiers is always above 600 after 1200 explore trials and does not show a trend of reduction until the end of the simulations.

4. CONCLUSION

This paper presents an extension of the rule combining (RC) technique from the previously binary mode XCS-RC to handle real values. Not only capable of fulfilling the task to get full correctness rate very quickly, RC minimizes the number of rules for both single-step and multi-step tasks. The success of XCS-RC in performing superior to a number of XCS variants, adopting its concepts to cope with continuous-valued input is a major improvement to the learning system.

RXCS-RC has been compared to XCSR using the real-valued multiplexer scenario. As a result, RXCS-RC shows a better performance in mapping the environment quicker and providing correct answers more often. Not only that, it also succeeds on maintaining a more compact population. In the future, the RC is expected to be able to learn to map environments having non-rectangular areas. It is also an obvious objective to get a combination of RXCS-RC with other machine learning methods. Furthermore, it is ready to be implemented in a more physical environment involving real agents and facing more challenges in learning.

5. REFERENCES

[1] N. Fredivianus, H. Prothmann, and H. Schmeck. Xcs revisited: A novel discovery component for the extended classifier system. In *Simulated Evolution and Learning*, Lecture Notes in Computer Science, pages 289–298. Springer Berlin / Heidelberg, 2010.

[2] S. W. Wilson. Generalization in the XCS classifier system. In J. R. Koza et al., editors, *Genetic Programming*, pages 665–674. Morgan Kaufmann, 1998.

[3] S. W. Wilson. Get real! xcs with continuous-valued inputs. In *Learning Classifier Systems, From Foundations to Applications*, pages 209–222, London, UK, 2000. Springer-Verlag.