Dynamically Adding Sensors to the XCS in Multistep Problems: A Sensor Tagging Approach

Liang-Yu Chen
Institute of Biomedical Engineering, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan, R.O.C..
lychen1211@cs.nctu.edu.tw

Po-Ming Lee
Institute of Computer Science and Engineering, Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan, R.O.C..
pmli@cs.nctu.edu.tw

Tzu-Chien Hsiao
Biomedical Electronics Translational Research Center and Biomimetic Systems Research Center, Institute of Biomedical Engineering, Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan, R.O.C..
labview@cs.nctu.edu.tw

ABSTRACT

Dynamically adding sensors to the Extended Classifier System (XCS) during its learning process in multistep problems has been demonstrated feasible by using messy coding (XCSm) and s-expressions (XCSL) as the representation of classifier conditions. XCSm and XCSL shown improved performance when new sensors were dynamically added to the agent of these systems in addition to the original available sensors during the learning process. However, these systems may suffer from overspecified problem and some logical operators (or clauses) could lead instability of the performance. Despite studies have suggested that these issues can be solved by appropriate parameter tuning, in our previous finding, we introduced a novel representation of classifier conditions for the XCS, named Sensory Tag (ST) (called XCS with ST condition, XCSSTC) to achieve the same goal as XCSm and XCSL but inherent most of the mechanisms of the XCS to solve those issues that the XCSm and XCSL encountered without any parameter tuning. The experiments of the proposed method were conducted in the multistep problems (i.e. Woods1 and Maze4). The results indicate that the XCSSTC is capable of being dynamic added additional sensors to improve performance during the learning process, and moreover, the XCSSTC shown a better performance in regard to learning speed than the other methods.

Categories and Subject Descriptors

F.1.1 [**Models of Computation**]: Genetics-Based Machine Learning, Learning Classifier Sysytems

Keywords

Learning Classifier Systems; XCS, Scalability; Machine Learning

1. INTRODUCTION

Learning classifier system (LCS) has become one of the research mainstreams in the field of intelligent system. Various LCSs have been proposed, the Extended Classifier System (XCS) is one of the most popular ones in the application domain. Dynamically adding sensors to the XCS during its learning process in multistep problems has been demonstrated feasible by using messy coding (XCSm) and s-expressions (XCSL) as the representation of classifier conditions [1-2]. However, the XCSm and XCSL are slower than the ordinary XCS, and moreover, the XCSm may suffer from overspecified problem, and furthermore, some logical operators (or clauses) could cause instability in XCSL's performance. We proposed an XCS with ST condition called XCSSTC to achieve the same goal as the XCSm and XCSL with a better performance in regard to learning speed, and without having those issues which XCSm and XCSL encountered [3], Its effectiveness has been verified in the Multiplexer (MUX) problem domain [4] and it can dynamically learn multiple problems[5]. The results indicate that the proposed XCSSTC method can learn the 135-bit MUX problem and proved the superiority of the XCS with code-fragment conditions (XCSCFC) in reusing building blocks of knowledge [6]. This paper validated that the performance of the XCSSTC in the multistep problem domains (i.e. Woods1 and Maze4).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
GECCO'15 Companion, July 11-15, 2015, Madrid, Spain
ACM 978-1-4503-3488-4/15/07.
http://dx.doi.org/10.1145/2739482.2764680

2. THE METHOD AND MATERIALS

2.1 ST as the Representation of Classifier Conditions in Multistep Problems

The method we proposed here is to use the HT to implement the concept of STs as the representation of classifier conditions. In the proposed XCSSTC, the ST of a sensor is taken as a key whereas its Sensory Value (SV) is taken as the value corresponds to the key. Each classifier has its own HT as the representation of classifier conditions. The HT includes all the (ST, SV) data pairs. The condition bit which is # in a classifier of an ordinary XCS is simply ignored from the HT for the classifiers in the XCSSTC.

2.1.1 Matching Classifiers in the XCSSTC

In the XCSSTC, a matching process decides whether a classifier is matched by enumerating all the SVs in the HT of the *cl*. The procedure of the XCSSTC to match a classifier *cl* is to compared to the corresponding bit position (by using the ST) in the input string and the absent (ST, SV) pairs are considered as "don't care" and hence ignored.

2.1.2 Covering in the XCSSTC

In the covering process of the XCSSTC, a random classifier whose condition matches the current environmental state *s* and each (ST, SV) pair in the HT has probability $P_\#$ to be taken as "don't care".

2.1.3 Rule Discovery in the XCSSTC

First selects two parent classifiers from the [A] based on their fitness and produces two offspring by the parents. Then the conditions of two offspring are crossed with probability χ. Uniform Crossover (UX) is used for the GA in the XCSSTC. After the crossover operation, each (ST, SV) pair has a probability μ to be removed in the mutation operation. The absent STs are considered as "don't care" attributes. Then, with probability μ, the XCSSTC randomly chooses an action for child.

2.1.4 Subsumption Deletion in the XCSSTC

In the XCSSTC, there are two steps to determine which classifier rule is more general than the other classifier rule that can subsume the other. The first step is to count the number of the (ST, SV) pairs in the HT. The number of the absent STs is equivalent to the number of "don't care" bit. This action determines whether the classifier is general or not efficiently. The second step is to compare the SVs of the classifiers corresponds to their STs. If cl_1 has a ST that cl_2 does not have, then cl_2 cannot subsumed into cl_1.

2.1.5 Encapsulating and Reusing the Learned Knowledge When Adding New Sensors to the XCSSTC

The agent learn from the environment with only four sensors and the learned population set was taken as the $[P]_{-1}$. Later, when adding the additional new four sensors to the XCSSTC, the covering process of the XCSSTC is then altered to a two-step procedure. The first step is to use the ordinary matching process to match the classifier rules in the $[P]_{-1}$ and pick out an experienced classifier ($exp > \theta_{sub}$) which has the highest fitness value. The second step is to apply the covering operation for the additional new STs for the classifier selected from the $[P]_{-1}$. For a probability of $P_\#$ the new (ST, SV) is not inserted into the classifier's HT. On the other hand, if there is no classifier matches the environmental input in the $[P]_{-1}$, the XCSSTC applies the ordinary covering operation to create a new classifier.

2.2 Experiment Design

The environment of multistep problems is a grid in which an agent is placed in the grid to find a food. The agent has a number of sensors to perceive the adjacent situation of corresponding cell in the environment. Each cell in the grid can be an obstacle ("O" with sensor codes 01), a food ("F" with sensor codes 01), or it can be empty ("*" with sensor codes 00). The agent can move into any of adjacent cell. If the adjacent cell is an obstacle, the agent unable to move into this cell and left in the original place; if the adjacent cell is empty then the agent can move into the cell; finally, if the adjacent cell is a food, the agent will move into the cell and receive a constant reward to end the problem. The experimental environments used here are Woods1 and Maze4. The experiment of the XCSSTC on each experiment was repeated for 30 times with a different random seed and all the reports are average result of the 30 runs.

3. RESULT

Figure 1 and Figure 2 show the effect of XCSSTC in Woods1 and Maze4. XCSSTC with reusing the $[P]_{-1}$ (241.66±141.47) is not significantly ($t(58) = 8.146$, $p > .05$) faster than XCSSTC without reusing the $[P]_{-1}$ (603.33±192.75) in Woods1 problem. XCSSTC with reusing the $[P]_{-1}$ (433.33±200.55) is significantly ($t(58) = 6.230$, $p < .05$) faster than XCSSTC without reusing the $[P]_{-1}$ (1108.33±547.88) in Maze4 problem.

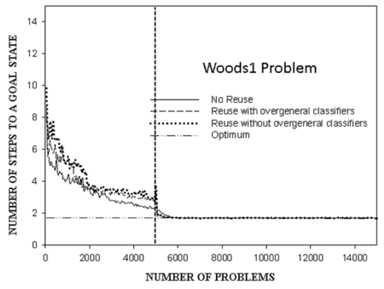

Figure 1 Result of XCSSTC by reusing the four cardinal sensors without overgeneral classifier for Woods1 problem.

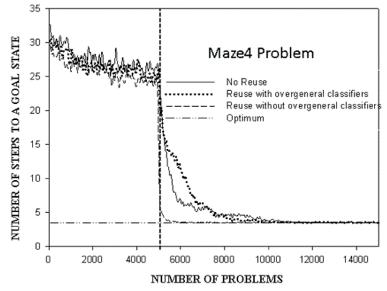

Figure 2 Result of XCSSTC by reusing the four cardinal sensors without overgeneral classifier for Maze4 problem

4. ACKNOWLEDGMENTS

This work was fully supported by Taiwan Ministry of Science and Technology under grant numbers MOST 103-2221-E-009-139. This work was also supported in part by the "Aim for the Top University Plan" of the National Chiao Tung University and Ministry of Education, Taiwan, R.O.C..

5. REFERENCES

[1] Lanzi, P. L. 1999. Extending the representation of classifier conditions part i: from binary to messy coding. In *Proceedings of the genetic and evolutionary computation conference*. 1, 337-344.

[2] Lanzi, P. L. and Perrucci, A. 1999. Extending the Representation of Classifier Conditions Part II: from messy coding to S-Expressions. In *Proceedings of the Proceedings of the genetic and evolutionary computation conference*. 1, 345-352.

[3] Chen, L.-Y., Lee, P.-M., Hsiao, T.-C. 2015. A Novel Representation of Classifier Conditions named sensory tag for the XCS in multistep problems. In *Proceedings of the The Genetic and Evolutionary Computation Conference*.

[3] Chen, L.-Y., Lee, P.-M., Hsiao, T.-C. 2015. A Sensor Tagging Approach For Reusing Building Blocks of Knowledge. In Learning Classifier Systems. *IEEE Congress on Evolutionary Computation*.

[5] Wu, Y.-M., Chen, L.-Y., Lee, P.-M., Hsiao, T.-C. 2015. Enable the XCS to Dynamically Learn Multiple Problems: A Sensor Tagging Approach. In *Proceedings of the The Genetic and Evolutionary Computation Conference*.

[6] Iqbal, M., Browne, W. N. and Zhang, M. 2012. Extracting and using building blocks of knowledge in learning classifier systems. In *Proceedings of the Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*. 863-870.