ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-761

Intuitive Action Set Formation in Learning Classifier Systems with Memory Registers

L. Simões and M.C. Schut and E. Haasdijk[1]

Abstract. An important design goal in Learning Classifier Systems (LCS) is to equally reinforce those classifiers which cause the level of reward supplied by the environment. In this paper, we propose a new method for action set formation in LCS. When applied to a Zeroth Level Classifier System with Memory registers (ZCSM), our method allows the distribution of rewards among classifiers which result in the same memory state, rather than those encoding the same memory update action.

1 INTRODUCTION

This paper introduces a new method for action set formation (*asf*) in Learning Classifier Systems, and tests it in partially observable environments requiring memory. The operation of *asf* is responsible for choosing the classifiers that will receive the reward supplied by the environment, for some performed action. When new classifiers are generated, the system has no way of knowing how good these are. Their strengths depend on the actions in the contexts under which they trigger, and on the other classifiers in the population with which they interact. As classifiers are added to the population, these are assigned an initial strength value. Then, by repeated usage, the strength update component will gradually converge towards a better estimate of their qualities. But since the system has to perform at the same time it is building its rule base, it is forced to act despite its uncertainty about the environment, and selecting from among an ever changing population of insufficiently tested classifiers. The method introduced here, *iasf*, eliminates some of the noise to which the quality estimation component is subjected, with the goal of improving system performance.

2 BACKGROUND

In the mid-1990s, Wilson [7] proposed ZCS as a simplification of Holland's original LCS [3]. Most importantly, he left out the message list which acted as memory in the original system. Thus, Wilson's models had no way of remembering previously encountered states and could not perform optimally in partially observable environments where an agent can find itself in a state that is indistinguishable from another state. However, the best action to undertake is not necessarily the same in both states. Wilson proposed [7] a solution for this problem in the form of memory registers to extend the classifiers. Cliff & Ross [2] follow this suggestion and implement ZCSM, extending ZCS with a memory mechanism. In their experiments they observed that ZCSM can efficiently exploit memory in partially observable environments.

Stone & Bull extensively compared ZCS to the more popular XCS in noisy, continuous-valued environments [6] and found that what makes XCS so good in deterministic environments (namely; its attempt to build a complete, maximally accurate and maximally general map of the payoff landscape) becomes a disadvantage as the level of noise in the environment increases. ZCS's partial map, focusing on high-rewarding niches in the payoff landscape then becomes an advantage. This suggests ZCS as an adaptive control mechanism in multi-step, partially observable, stochastic real-world problems.

3 INTUITIVE ACTION SET FORMATION

ZCS works on a *population* P of rules which together present a solution to the problem with which the system is faced. As it interacts with the environment, the system is triggered on reception of a sensory input. A *match set* M is then formed with all the rules in the population matching that input. From this set, a classifier is chosen by proportionate selection based on its strength, and its action is executed. With memory added as described in [2], rules prescribe an external action as well as a modification of the memory bits.

It can be argued that the core of ZCS lies in the next, reinforcement stage, as it is responsible for incrementally learning the quality of the rules in the population, which will in turn determine the system's behaviour.

The action set A includes those rules in M that advocated the same action as the chosen classifier. The rules in this action set share in the reward that results from the selected action (with the rationale that choosing any of those rules would have had the same effect). Rules in M that advocate a different action are penalised.

Traditionally, A consists of those rules in M that match on a *bitwise comparison with the action-part* of the chosen classifier. Now, consider ZCSM, where operators on the memory state are added to the action part of the rules. Suppose, then, a situation where the memory state was `01`, and remains the same after execution of some chosen classifier c, which advocated[2] `[0#]`. Traditional action set formation would then have A include only those classifiers from M advocating this same *memory operation* ("set the first memory register to 0") as well as the same external action as the chosen classifier. However, all of the internal actions {`##`,`#1`,`01`} would result in exactly the same internal state. Not only would the system not reward any classifier in M having one of those internal actions (and the same external action) as the chosen classifier, it would actually penalise them. This seems to conflict with ZCS's goal of equally rewarding those classifiers which would cause the same level of reward supplied by the environment.

[1] Department of Computer Science, Faculty of Sciences, VU University, Amsterdam, The Netherlands, email: {lfms, mc.schut, e.haasdijk}@few.vu.nl

[2] Disregarding the external output for simplicity.

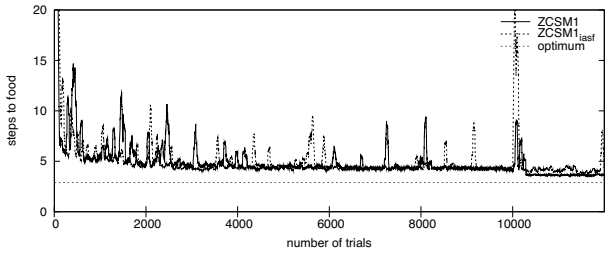

Figure 1. Performance comparison in `woods101` with 1 memory bit.

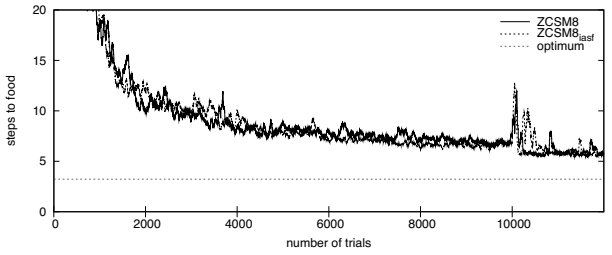

Figure 2. Performance comparison in `woods102` with 8 memory bits.

This realisation prompted us to introduce a new variant of Cliff & Ross' classifier system, ZCSM$_{iasf}$, which compares classifiers based on the memory state which would *result* from their activation, rather than based on the memory *operation*. In this more intuitive scheme, any rule in M that prescribes the same external action as c and an internal action that leads to the same memory state (i.e., one of {`##`,`#1`,`0#`,`01`}) is included in A.

4 EXPERIMENTAL ANALYSIS

Experimental Design and Setup – To compare the performance of *iasf* against regular action set formation, we conducted a series of experiments in the well-known `woods101` and `102` environments [2, 5]. These are mazes where paths towards food locations must be learned; both mazes contain indistinguishable locations where the sensory information (i.e., the layout of the perceivable cells) is identical but the appropriate action differs. To tackle such situations, the agent's controller requires memory to be able to choose the correct action; only reacting to sensory information cannot suffice.

An experiment consists of 10,000 trials where, starting from a random location in the maze, the agent must reach the food. If the agent moves into the cell with food, it receives a reward from the environment and the next trial commences: the food is replaced and the agent is randomly relocated. The agent can see the directly adjacent cells and uses that information to decide on an action—where to move next. Following Bull & Hurst's suggestion, the system is then further tested for an additional 2,000 trials where "the Genetic Algorithm is switched off, reinforcement occurs as usual, and an action selection scheme is used which deterministically picks the action with the largest total fitness in M" [1]. Performance is measured as the moving average over the previous 50 trials of the number of steps it took to reach the food on each trial. See [7, 2] for more detailed descriptions of the experimental setup.

We performed experiments with a memory size of 1 in `woods101` and 8 in `woods102` with Wilson's default parameter set for ZCS [7]. Given the more demanding characteristics of `woods102`, we used a larger population size (N = 2000) there.

Results – Figures 1 and 2 show the results of experiments averaged over 30 runs; the lighter horizontal line shows the optimal average performance for each environment (2.9 steps for `woods101` and 3.23 for `woods102` [5]). The horizontal axes show the number of trials into the experiment.

Analysis – Although the change in *asf* technique is an intuitive one and one that fulfils the LCS design goal of equal credit assignment to the classifiers producing the level of reward coming from the environment, no benefit in performance can be gleaned from the results of our experiments. In both cases, ZCSM$_{iasf}$ performed at substantially the same level as traditional ZCSM; only in `woods102` can we see some slight –not statistically significant– improvement. Because this is the more challenging of the two environments [5], this may indicate that performance in more complex environments and tasks can benefit from *iasf*, but this remains an issue for further investigation.

5 CONCLUSIONS

We have extended the way action sets are formed in classifier systems with memory registers, taking them closer to the design goal of equal credit assignment to the classifiers whose actions cause the level of reward supplied by the environment. We have validated our extension experimentally in partially observable environments using the Zeroth Level Classifier System.

The environments on which experiments were performed are well-known in the existing literature on the subject. The experiments showed no significant improvement in performance. We require further investigation to see whether such improvement does occur in more complex environments. Still, the current results can be considered valuable since the new method is more in line with the general design goal of equal credit assignment than the traditional method.

In stochastic environments, where the ZCS algorithm has previously shown to outperform the more widely known XCS [6], rule quality estimation can be expected to take on a more significant role, which leads us to think that our extension will provide more significant benefits in partially observable instances of those problems. Again, further investigations are required to validate this assumption.

REFERENCES

[1] Larry Bull and Jacob Hurst, 'ZCS redux', *Evolutionary Computation*, **10**(2), 185–205, (2002).
[2] Dave Cliff and Susi Ross, 'Adding temporary memory to zcs', *Adaptive Behavior*, **3**(2), 101–150, (1994).
[3] John H. Holland, 'Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems', in *Machine learning, an artificial intelligence approach*, eds., R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, volume 2, Morgan Kaufmann, (1986).
[4] Pier Luca Lanzi, 'An analysis of the memory mechanism of XCSM', in *Genetic Programming 1998: Proceedings of the Third Annual Conference*, eds., John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, pp. 643–651, San Francisco, CA, USA, (22-25 July 1998). Morgan Kaufmann.
[5] Pier Luca Lanzi and Stewart W. Wilson, 'Toward optimal classifier system performance in non-markov environments', *Evolutionary Computation*, **8**(4), 393–418, (2000).
[6] Christopher Stone and Larry Bull, 'Comparing XCS and ZCS on noisy continuous-valued environments', Technical Report UWELCSG05-002, Learning Classifier Systems Group, University of the West of England, Bristol, UK, (2005).
[7] Stewart W. Wilson, 'ZCS: A zeroth level classifier system', *Evolutionary Computation*, **2**(1), 1–18, (1994).