Evol. Intel. (2015) 8:51–53
DOI 10.1007/s12065-015-0131-0

EDITORIAL

Special issue on the 20th anniversary of XCS

Tim Kovacs[1] · Muhammad Iqbal[2] · Kamran Shafi[3] · Ryan Urbanowicz[4]

Received: 29 March 2015 / Accepted: 30 March 2015 / Published online: 12 April 2015

This issue is dedicated to Stewart Wilson both in recognition of his achievements and in appreciation for his guidance of the field. Not only has his research transformed the field, but his support, encouragement, and advice have been invaluable to many of us individually and to the community as a whole.

Learning Classifier Systems (LCS) are rule-based machine learning algorithms introduced by Holland (see [2, 4–6]) that learn a population of IF-THEN rules that specify "IF x happens THEN do (or predict) y". In 1995 Stewart Wilson, already a leading LCS researcher, published a paper entitled "Classifier Fitness based on Accuracy" [16] that introduced the XCS algorithm. This paper proved a turning point for the field as XCS and its derivatives rapidly became the main focus of LCS research, and continue to be so today. We organised this special issue of *Evolutionary Intelligence* to mark the 20th anniversary of this landmark paper, and to serve as post-proceedings for IWLCS 2014: the Seventeenth International Workshop on Learning Classifier Systems. The annual IWLCS meetings are the yearly highlight of the LCS calendar and have no doubt contributed to the strong sense of community in the LCS field. For 2015 IWLCS has been rebranded as the more descriptive IWERML: the International Workshop on Evolutionary Rule-based Machine Learning.

In 1994, the year before introducing XCS, Wilson introduced ZCS, the "Zeroth-order Classifier System" [15]. Wilson felt the growing complexity of the LCS concept was hindering progress and ZCS was an attempt to strip the LCS down to its essentials while retaining a working system. XCS built on the minimalist ZCS with two radical changes: a switch to accuracy-based fitness and, based on work by Booker [1], the addition of a niche Genetic Algorithm (GA) that strongly favours general rules. This combination was a hit: the niche GA favours general rules but accuracy-based fitness insists strictly that they make good predictions. This combination drives XCS to learn rules that are as general as possible while remaining accurate. Bull's article in this issue provides much more on the history of LCS prior to and following XCS [2].

The famous "accuracy-based fitness" of XCS needs some explanation. The earlier ZCS featured "strength-based fitness", in which the fitness of a rule in the GA was derived from its strength, a measure of the amount of reward the rule received. Consequently, fit rules in ZCS are those that receive a lot of reward. In contrast, XCS rules are fit if they make consistently accurate predictions about the reward they receive. This has the counterintuitive consequence that XCS rules that consistently take a bad action can be fit, since they are consistent. However, this poses no problem when it comes time for XCS to choose an action, since that can be done using the magnitude of the reward predicted by each rule. Why does XCS retain rules whose action it does not use? This "complete map" of the state/

✉ Muhammad Iqbal
muhammad.iqbal@ecs.vuw.ac.nz

Tim Kovacs
Tim.Kovacs@bristol.ac.uk

Kamran Shafi
k.shafi@adfa.edu.au

Ryan Urbanowicz
ryan.j.urbanowicz@dartmouth.edu

[1] Department of Computer Science, University of Bristol, Bristol, UK

[2] School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand

[3] School of Engineering and Information Technology, University of New South Wales, Canberra, Australia

[4] Department of Genetics within Molecular and Cellular Biology, Dartmouth College, Hanover, NH, USA

action space allows XCS to build, in the terminology of reinforcement learning (RL), a long-term value function over the state/action space. In fact, XCS implements a rule-based form of the famous RL Q-learning algorithm [14]. The fact that the map is complete allows XCS to learn about every state/action combination, which is an advantage in RL tasks given the typical uncertainty about which action is best for a given state. There is more to be said on the comparison of strength and accuracy-based fitness for RL than space allows here [3, 7].

Some tasks, however, are better modelled as supervised learning, and an important derivative of XCS called UCS (for sUpervised Classifier System) [12] fits this niche. One feature of UCS is that it learns only rules whose actions it uses, rather than a complete map, so its rule population is smaller than XCS's. A line of research has sought to retain XCS's RL updates, but to eliminate rules that are not needed for decision-making. Nakata et al. [9] introduced XCSAM (XCS with Adaptive action Mapping), which estimates how often each rule takes an optimal action and incorporates that estimate into the rule's fitness. In this issue, Nakata et al. [11] demonstrate that the fitness-based rule selection XCSAM inherited from XCS can, in some problems, result in the population retaining large numbers of suboptimal rules. They then introduce a new variant of XCSAM with a new selection strategy which results in smaller population sizes.

Returning to supervised learning, Urbanowicz and Moore were motivated by problems in bioinformatics and genetic epidemiology to further optimise LCS algorithms for complex, real-world, noisy and heterogeneous problems. This lead to the development of an algorithm called ExSTraCS, which builds on the XCS/UCS foundation. In this issue [13] they introduce ExSTraCS 2.0 along with methods to greatly improve LCS scalability, which is an important consideration given the run-times they typically require and the current trend in machine learning toward big data. They go on to demonstrate that ExSTraCS 2.0 is able to solve the extremely complex 135-bit multiplexer benchmark problem "directly" for the first time ever reported in the literature. This work also provides the first complete specification of ExSTraCS.

XCS is still being extended today because it is a flexible framework built around the core ideas of accuracy-based fitness and genetic generalization. Much research has been devoted to piecewise function approximation with a variant called XCSF [17], in which each rule computes a real-valued prediction as a function f of its input x: "IF this rule applies to input x THEN output $f(x)$". Lanzi and Loiacono's work in this issue [8] is a good example of XCS's flexibility as it replaces the earlier functions used in XCSF with tile coding. Although tile coding is a popular function approximation method in RL its success depends on its parameterisation, and tile coding itself is not adaptive. In contrast, Lanzi and Loiacono's hybrid LCS/tile coding system is able to evolve appropriate tile coding approximators for various subspaces of the function being approximated. In this issue they demonstrate that their hybrid system is able to cope with discontinuous RL tasks designed to challenge tile coding.

Nakata et al.'s [10] XCS-SL (XCS for Sequence Labeling) is another example of XCS's flexibility. In some tasks it is useful to augment the current input to the learner with previous inputs. For example, a maze might contain two T-junctions that require the agent to move in different directions. Although the T-junctions might look identical, there may be some feature nearby which disambiguates them, e.g., a crack in the wall near one junction. If the learner remembers previous inputs it may be able to find associations between inputs at different times that help it make decisions. A number of LCS algorithms incorporate memory, but the XCS-SL work in this issue is the first to address sequence labelling, rather than RL tasks. Sequence labelling is a paradigm in which each input must be classified, but, unlike most supervised learning, there is a sequential relationship between inputs. Part of speech tagging is a good example, in which each word in a sentence must be labeled as a noun, verb, etc., and previous (or later) words can be useful in labelling the current one. XCS-SL addresses sequence learning by adding a variable-length memory window to each rule. Evolution optimises window size, and XCS learns to generalize over memories that are not needed. In other words, XCS's key innovation of accuracy-based fitness and genetic generalisation is repurposed in XCS-SL to accurately generalize over memories. What is more, this required only minimal changes to the rule representation and genetic operators. It is a good algorithm that is still finding new uses after nearly 20 years.

We think the papers in this special issue are a good snapshot of the state of XCS research approaching its 20th anniversary and we are looking forward to the next 20 years!

References

1. Booker LB (1989) Triggered rule discovery in classifier systems. In: Schaffer JD (ed) Proceedings of the 3rd international conference on genetic algorithms (ICGA89). Morgan Kaufmann, pp 265–274
2. Bull L (2015) A brief history of learning classifier systems: from CS-1 to XCS and its variants. Evol Intell
3. Bull L, Hurst J (2002) ZCS redux. Evol Comput 10(2):185–205
4. Holland JH (1976) Adaptation. In: Rosen R, Snell FM (eds) Progress in theoretical biology. Plenum, New York

5. Holland JH (1980) Adaptive algorithms for discovering and using general patterns in growing knowledge bases. Int J Policy Anal Inf Syst 4(3):245–268
6. Holland JH, Reitman JS (1978) Cognitive systems based on adaptive algorithms. In: Waterman DA, Hayes-Roth F (eds) Pattern-directed Inference Systems. Academic Press, New York. Reprinted In: Evolutionary computation. The Fossil Record. Fogel DB (ed) IEEE Press, 1998. ISBN: 0-7803-3481-7
7. Kovacs T (2004) Strength or accuracy: credit assignment in learning classifier systems. Springer, London
8. Lanzi PL, Loiacono D (2015) XCSF with tile coding in discontinuous action-value landscapes. Evol Intell
9. Nakata M, Lanzi PL, Takadama K (2012) XCS with adaptive action mapping. The ninth international conference on simulated evolution and learning (SEAL 2012), pp 256–265
10. Nakata M, Kovacs T, Takadama K (2015) XCS-SL: a rule-based genetic learning system for sequence labeling. Evol Intell
11. Nakata M, Lanzi PL, Takadama K (2015) Rule reduction by selection strategy in XCS with adaptive action map. Evol Intell
12. Orriols-Puig A, Bernadó-Mansilla E (2008) Revisiting UCS: description, fitness sharing, and comparison with XCS. In: Bacardit J, Bernadó-Mansilla E, Butz M, Kovacs T, Llorà X, Takadama K (eds) Learning classifier systems. 10th and 11th international workshops (2006–2007). Lecture notes in computer science, vol 4998/2008. Springer, pp 96–111
13. Urbanowicz RJ, Moore JH (2015) ExSTraCS 2.0: description and evaluation of a scalable learning classifier system. Evol Intell
14. Watkins C (1989) Learning from delayed rewards. Ph.D. thesis, Cambridge University
15. Wilson SW (1994) ZCS: a zeroth level classifier system. Evol Comput 2(1):1–18
16. Wilson SW (1995) Classifier fitness based on accuracy. Evol Comput 3(2):149–175
17. Wilson SW (2001) Function approximation with a classifier system. In: Spector L, Goodman ED, Wu A, Langdon W, Voigt HM, Gen M, Sen S, Dorigo M, Pezeshk S, Garzon MH, Burke E (eds) In: Proceedings of the genetic and evolutionary computation conference (GECCO-2001). Morgan Kaufmann, pp 974–981