Learning Classifier Systems: New Models, Successful Applications

John H. Holmes[a] Pier Luca Lanzi[b] Wolfgang Stolzmann[c] Stewart W. Wilson[d]

[a] Center for Clinical Epidemiology and Biostatistics
University of Pennsylvania, School of Medicine
Philadelphia, PA 19104 USA
jholmes@cceb.med.upenn.edu

[b] Artificial Intelligence and Robotics Laboratory
Dipartimento di Elettronica e Informazione
Politecnico di Milano, Italy
pierluca.lanzi@polimi.it

[c] DaimlerChrysler AG
Research and Technology, Berlin, Germay
Wolfgang.Stolzmann@daimlerchrysler.com

[d] Department of General Engineering
The University of Illinois, Urbana-Champaign, USA
Prediction Dynamics, Concord, USA
wilson@prediction-dynamics.com

Rules are an accepted means of representing knowledge for virtually every domain. Traditional machine learning methods derive rules by exploring sets of examples using statistical or information theoretic techniques. Alternatively, rules can be discovered through methods of Evolutionary Computation such as Genetic Algorithms (GA) and Learning Classifier Systems (LCS).

In recent years, new models of Learning Classifier Systems have been developed which have resulted in successful applications in a wide variety of domains (e.g., autonomous robotics, classification, knowledge discovery, modeling). These models have led to a resurgence of this area which for a certain period appeared almost at a dead end. This paper overviews the recent developments in learning classifier systems research, the new models, and the most interesting applications, suggesting some of the most relevant future research directions.

1. Introduction

Rules are an accepted means of representing knowledge for virtually every domain. For instance, they can be used in classification, prediction, and modeling. Traditional machine learning methods derive rules from sets of examples by means of statistical or information theoretic techniques. Alternatively, rules can be *discovered* through methods of Evolutionary Computation such as *genetic algorithms* and *learning classifier systems* [1].

Learning Classifier Systems (LCS) are a *machine learning paradigm* introduced by John H. Holland [2], the father of genetic algorithms. They exploit *evolutionary computation* and *reinforcement learning* to develop a set of condition-action rules (i.e., the classifiers) which represent a target task that the system has learned from on-line experience.

Learning classifier systems have been in existence for more than twenty years. While there was considerable research in the 1980's, the field began to wane as the decade closed. In the early 1990's, learning classifier systems seemed too complicated to be studied, with few successful applications reported. In the mid 1990's the field appeared almost at a dead end. But, during the last five years, new models have been developed

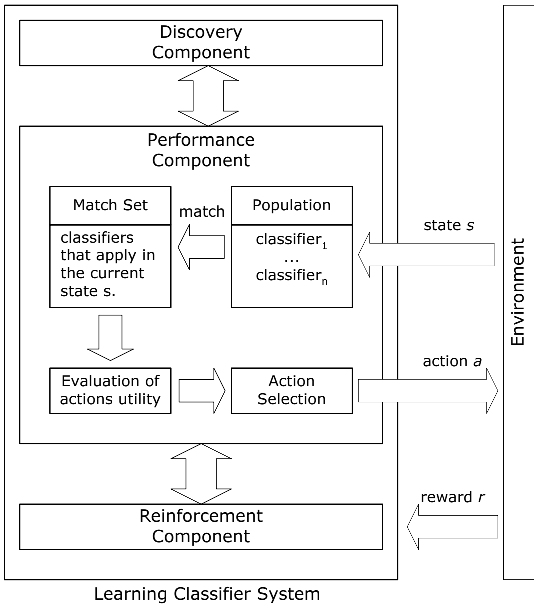

Figure 1. The learning classifier system and the environment

and new applications have been presented which caused a great resurgence of this area.

In this paper we present an overview of these recent developments and discuss the most interesting applications reported and the most promising future research directions. This paper is aimed at readers who do not know much about learning classifier systems and should be considered a sort of *pocket guide* to the recent developments in this area. Thus, we keep discussions as simple as possible leaving out some technical details. Furthermore, we focus on a few quite recent papers and do not tackle the whole history of the area (for that kind of survey we refer the interested reader to the book *Learning Classifier Systems: From Foundations to Applications* [1]).

2. What is a Learning Classifier System?

A learning classifier system learns by *interacting* with an *environment* from which it receives feedback in the form of numerical *reward*. Learning is achieved by trying to maximize the amount of reward received. There are many models of LCSs and therefore also many ways of defining what a learning classifier system is (e.g., Holland et al. [3]). Nevertheless, all LCSs models, more or less, comprise *four* main components (Figure 1): a finite *population* of condition-action rules, called classifiers, that represents the current knowledge of the system; the *performance* component, which governs the interaction with the environment; the *reinforcement* component (also called *credit assignment* component [2]), which distributes the *reward* received from the environment to the classifiers accountable for the rewards obtained; the *discovery* component, which is responsible for discovering better rules and improving existing ones through a genetic algorithm.

Classifiers have two associated measures: the *prediction* [4] (also called *strength* [2]), and the *fitness*.[1] Prediction estimates the classifier utility in terms of the amount of reward that the system will receive if the classifier is used. Fitness estimates the "quality" of the information about the problem that the classifier conveys, and it is exploited by the discovery component to guide evolution: a high fitness means that the classifier conveys "interesting" information about the problem and therefore it should "reproduce" more through the genetic algorithm; a low fitness means that the classifier conveys little or no "interesting" information about the problem and therefore should "reproduce" less.

On each *discrete* time step (Figure 1), the system receives as input the current *state* of the environment s and builds a *match set* containing the classifiers in the population whose condition *matches* the state s. Then, the system evaluates the utility of the actions appearing in the match set; an action a is selected from those in the match set according to a certain criterion, and sent to the environment to be performed. Depending on the current state s and on the consequences of action a the system eventually receives a reward

[1] In Holland's original framework the *strength* is also used as a measure of the classifier *fitness*, i.e., both strength and fitness are represented by the same parameter. Here we prefer to separate these concepts since recent models use distinct parameters for strength and fitness.

r. The *reinforcement* component distributes r among the rules accountable of the incoming rewards. This can be either implemented with an algorithm specifically designed for learning classifier systems (e.g., the *bucket brigade algorithm* in [2]) or with an algorithm inspired by traditional reinforcement learning methods (e.g., the modification of Q-learning in [4]).

On a regular basis, the *discovery* component, i.e., the genetic algorithm, randomly selects two classifiers from the population with probability proportional to their fitnesses; it applies crossover (i.e., sexual recombination) and mutation generating two new classifiers. These are inserted in the population while two others are deleted in order to keep the population size constant.

While genetic algorithms represent the most common way of implementing the discovery component, over the years some "non-evolutionary" implementations have been proposed (e.g., [5,6]). Such alternative implementations, based on different search heuristics, have proved effective in some domains (e.g., [7–9]). However, genetic algorithms are still regarded as the most general way of guiding rule discovery in learning classifier systems, also from a theoretical standpoint (e.g., [10]).

The environment defines the target task. For instance, in autonomous robotics [6] the *environment* corresponds *roughly* to the robot's physical surroundings and the goal of learning is to learn a certain behavior. In classification problems [11,12], the environment is a *training* set of preclassified examples; each example is described by a vector of attributes and a *class* label; the goal of learning is to evolve rules that can be used to classify previously unseen examples with high accuracy. Finally, in *Computational Economics*, the environment represents a market and the goal of learning is to make profits.

3. Why Learning Classifier Systems?

Three arguments are usually presented to support learning classifier systems: adaptivity, generalization, and scalability.

Adaptivity. Learning classifier systems have always been viewed as *adaptive* systems capable of *on-line* learning in rapidly changing situations by exploiting their genetic algorithm. However, only in recent years has true evidence of this important characteristic been reported. For instance, Holmes [11] applied an LCS, EpiCS, to *epidemiologic surveillance* and demonstrated that LCSs can be effective both for predicting and for describing a rapidly evolving phenomenon, such as the occurrence of a certain epidemic disease. Furthermore, in Computational Economics [13] LCSs are successfully employed for modeling the behavior of buyers and sellers in artificial stock markets [14–16] where agents must be able (i) to adapt to rapidly changing market situations, and (ii) to develop successful strategies against opponents. Finally, we observe that in learning classifier systems adaptivity is effective over different time scales: in [11] it works on the data collected during weeks or months, in [16] on the data collected daily, while in other cases (e.g., [14,15]) the data arrive in real-time.

Generalization is an important feature of learning systems since a system which generalizes properly can: (i) represent in a compact form what it has learned and can (ii) apply what it has learned to previously unseen situations. Generalization is important in many application domains. For instance, in Knowledge Discovery and Data Mining generalization capabilities are essential for developing compact descriptions of the phenomena analyzed which also apply with high accuracy to previously unseen data.

In learning classifier systems generalization is achieved through the evolution of *general* rules which apply in many situations, i.e., which match many environmental states. As with adaptivity, generalization has been studied in LCSs since the early days, but evidence for it was rarely reported (see [17] for an overview of early results). In contrast, recent LCS models have demonstrated powerful generalization capabilities. For instance, Wilson [4] showed that an LCS, XCS, can evolve *minimal* sets of *maximally general* classifiers which accurately describe the solution for a certain learning task. Furthermore, such results hold for various rule syntaxes [18,12], suggesting that the generalization capabilities of LCSs are likely to be independent of rule syntax, as also discussed by Lanzi [10] from a more theoretical perspective.

Scalability. One important issue for LCS research is that of *scalability*, i.e., how rapidly does the learning time or the system size grow as the problem complexity increases? Although scalability is important for LCSs, there are no clear answers so far. This is for two main reasons. First, in LCSs the genetic algorithm does not readily allow evaluation of the computational complexity of LCSs. Furthermore, most of the LCSs applications reported until the recent years dealt with problems of limited complexity and thus they did not raise particular concerns about scalability. There are however two significant exceptions.

Finding that a single LCS was not sufficient to train autonomous robots to perform complex tasks, Dorigo and Colombetti [19] developed an entire methodology, namely BAT, to *incrementally* train autonomous robots to perform complex tasks through a hierarchy of LCSs (see [19] for details). Separately, Wilson [20] proposed the *first* theoretical hypothesis regarding the complexity of learning in LCSs. Specifically, Wilson suggested that (with respect to his XCS classifier system [4]) the complexity of the learning process grows as a *low order polynomial* of the complexity of the problem, *not* with the complexity of the learning space as in other techniques (e.g., neural networks). So far Wilson's hypothesis has only been tested on a limited number of experiments on complex Boolean functions. However, it is worth noting that such results were basically impossible with earlier LCS models.

4. Which Models?

For many years, "LCS" just stood for "Holland's LCS" and most research was focused on trying to solve the problems that the original model had. Today, "LCS" identifies the *paradigm* proposed by Holland while there are different models in which most of the problems of Holland's framework have been solved. Among others, three models introduced in the recent years appear particularly promising.

Wilson's XCS is the culmination of the research Wilson started in 1987 for developing a *new* LCS in which Holland's ideas could work within a *simpler* model [21]. In these years, Wilson's research resulted in important models (NEWBOOLE, ZCS [17]) which finally led to XCS, currently the most studied and most promising research direction in LCS.

XCS keeps all the main ideas of Holland's model while it introduces some fundamental changes. First, in XCS the fitness of classifiers depends on the accuracy of a classifier's prediction and *not* on the prediction itself as in Holland's model. Thus, in XCS, a classifier is considered "useful" if it conveys *accurate* information about the problem. Second, in XCS the discovery component does not act on the whole population (as usually happened in most LCS models) but on subsets of classifiers which apply to the same situations, i.e., in XCS the discovery component acts in *environmental niches* not in the whole population. Finally, XCS's architecture is much neater than that of previous models; accordingly XCS is easier to study and analyze since it puts more in evidence the role of the various LCS components.

Although these might appear small changes, overall they resulted in an LCS, XCS, which (i) can tackle complex learning problems, practically unsolvable with former models; (ii) produces solutions that are extremely compact, which consist of a small number of *maximally general* and *accurate* classifiers [20]; (iii) can use different kinds of representations [18]; (iv) can be easily studied and analyzed so as to allow the development of formal models and the building of strong connections with more traditional machine learning [22,10]. For these reasons, XCS represents an important (probably, the most important) breakthrough of LCS research in the last ten years. In fact, XCS is currently the most studied and the most applied LCS, and surely represents a fundamental research direction for the next years.

Stolzmann's Anticipatory Classifier System (ACS) [6] differs greatly from other LCS models in that ACS learns *not only* how to perform a certain task, but also an internal model of the dynamics of the task (i.e., the environment).

In ACS classifiers are not simple condition-action rules but they are extended by an *effect* part. The effect-part of a classifier is used to anticipate the environmental state which results from the execution of the classifier action. The model of the environment can be learned latently, i.e., without any environmental reward, because the fitness of the classifiers depends on the accuracy of the anticipation. The classifier fitness is high if the next state is anticipated correctly while it is low if the anticipation is wrong. Besides Genetic Algorithms an Anticipatory Learning Process (ALP) is used for rule discovery which directly learns from the changes in the environment. The ALP is a further development of a psychological learning theory, called anticipatory behavioral control [6]. ACS forms explicit condition-action-effect classifiers with a generalization capability in the classifier conditions. This leads to an internal model of the environment which consists of a minimal set of classifiers. The internal model can be used in many applications: (i) for mental acting and lookahead-planning to improve learning, (ii) for action-planning and goal-directed planning in the absence of environmental reward, and many more [6].

Holmes' EpiCS [11] is an LCS specialized for classification and knowledge discovery tasks. It was developed from NEWBOOLE to meet the demands of epidemiologic data. EpiCS's distinctive features include: (i) techniques for controlling over- and under-generalization of data, (ii) the use of differential negative reinforcement of false positive and false negative errors in classification, and (iii) a methodology for determining risk as a measure of classification. All of these features have led to the successful use of EpiCS in knowledge discovery applications to actual clinical databases of various sizes and levels of complexity. EpiCS was able to (i) derive models that identified features that were associated with outcomes such as appropriate child restraint in automobiles, (ii) occupational cancer (simulations), and (iii) head injury to children involved in automobile crashes (see [23] for an overview). Therefore, EpiCS appears to be a successful approach to applying evolutionary computation to the realm of knowledge discovery in databases.

5. Which Applications?

LCSs have been applied in many domains [1]. However, most of the results reported fall into three main areas.

Autonomous robotics has always been an important testbed for LCSs, and it is probably the single research area where LCSs have been applied most during the last decade. The most extensive research in this area was carried out by Dorigo and Colombetti [19] with other colleagues. In the years from 1990 until 1998 they applied different versions of their parallel LCS, Alecsys, to a variety of robotics applications reporting interesting results. Apart from Dorigo and Colombetti, other interesting experiments have been reported (see [1]). Interestingly, few of these results employ the newer models (e.g., [7]), suggesting that autonomous robotics will be an important future challenge for more recent LCS models.

Knowledge Discovery is the area in which new LCS models have been applied most. In knowledge discovery applications, LCSs are used to extract *compact* descriptions of *interesting* phenomena described by multidimensional data. Holmes [11] applied his LCS, EpiCS, to extract interesting knowledge from different medical domains (see [23] for an extensive summary). Saxon and Barry [24] applied XCS to a set of known machine learning problems demonstrating that XCS is able to produce a classification performance and rule set which exceeds the performance of most current Machine Learning techniques. Wilson [12] applied XCS extended for integer inputs, XCSI, to the Wisconsin Breast Cancer dataset showing that XCSI performed as well as state-of-the-art machine learning algorithms while producing solutions (i) that are easily interpretable and (ii) suggest interesting dependencies on one or a few attributes. The recent results and the amount of ongoing work in this particular area suggest that knowledge discovery applications may represent an important breakthrough for LCSs, taking them well beyond the Evolutionary Computation community.

In **Computational Economics** [13], LCSs are used to model adaptive agents in artificial stock markets (e.g., Brian Arthur et al. [14], LeBaron et al. [15], Mitlöhner [25]). Bull [26] applied an LCS to a simulated "Continuous Double-Auction" (CDA) market. Recently, Schulenburg and Ross [27] employed LCSs to model the behavior of agents trading risk free bonds and risky assets in a stock market environment. Computational Economics is probably the area in which Holland's original model is *still* applied most whereas recent models (e.g., XCS, EpiCS) have not been tried yet. Thus we believe that this might become an important testbed for new LCS models (e.g., XCS, EpiCS) to challenge Holland's model.

Apart from these three areas, there are many other interesting applications of LCSs which cannot be discussed here. Thus, we refer the interested reader to [1] for further references.

6. Where to from here?

For a certain period, Learning Classifier Systems appeared almost at a dead end. But the development of new LCS models, successful in many domains, has led to a resurgence of this area during the last five years. Overall, the recent results represent probably the most significant advances in LCS research to occur so far. However, most work is still to be done; there are many interesting research directions to be explored, and many open challenges. Some of these are discussed below.

Generalization. Wilson's *Generalization Hypothesis* [28] suggest some principles that guide generalization in XCS. But at this time only a few works exist (e.g., [29,30]) that investigate generalization in XCS from a theoretical point of view. Thus important questions remain open, e.g.: What triggers the evolution of accurate and maximally general classifiers? What are the system requirements that insure effective generalization? What are the difficult classes of problems? Although, [28] provide some directions to look for these answers more work is needed to get definitive results within a more formal framework.

There is also an open debate concerning the use of genetic algorithms in learning classifier systems. Some researchers (e.g., [10,28]) look at genetic algorithms as the best search method for implementing generalization in LCSs while others (e.g., [6,8,9]) suggest that different search heuristics might be equally or even more effective than genetic algorithms. Thus, the question is: should we look for alternative ways to implement generalization? Or are genetic algorithms the best solution?

Scalability. Generalization is directly related to *scalability*. A system that can generalize properly can learn complex tasks *rapidly* and with *limited memory* resources. In [28] Wilson provided some formulas to relate the complexity of the learning tasks (namely, boolean multiplexer problems) to the computational time and the number of classifiers required for solving them. But those formulas apply to specific learning tasks. It would be important to extend Wilson's initial results to a wider class of tasks.

Anticipations. Learning classifier systems that can predict (i.e., anticipate) the effect of their actions are relevant in many application areas, e.g.: in autonomous robotics for planning, in financial application for stock market prediction. Current models of anticipatory classifier systems are mainly based on the work of Stolzmann [6] which is inspired to the anticipatory behavior in animals. But are there other feasible approaches for adding anticipations to LCSs? Might *classifier corporations* [31] provide another way to implement anticipatory behavior?

There are many researchers we could not mention in this paper, other models (e.g., extensions of XCS for non-Markov problems [32], the Pittsburgh classifier systems [33]) we could not discuss, and many results that we could not include. For further references and inspiration we suggest the recent book *Learning Classifier Systems: From Foundations to Applications* [1] or the forthcoming [34].

REFERENCES

1. P. L. Lanzi, W. Stolzmann, S. W. Wilson (Eds.), Learning Classifier Systems: From Foundations to Applications, Vol. 1813 of LNAI, Springer-Verlag, Berlin, 2000.
2. J. H. Holland, Escaping Brittleness: The possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems, in: M. et al. (Ed.), Machine learning, an artificial intelligence approach. Volume II, Morgan Kaufmann, 1986, Ch. 20, pp. 593–623.
3. J. H. Holland, L. B. Booker, M. Colombetti, M. Dorigo, D. E. Goldberg, S. Forrest, R. L. Riolo, R. E. Smith, P. L. Lanzi, W. Stolzmann, S. W. Wilson, What is a Learning Classifier System?, in: Lanzi et al. [1], pp. 3–32.
4. S. W. Wilson, Classifier Fitness Based on Accuracy, Evolutionary Computation 3 (2) (1995) 149–175.
5. R. L. Riolo, Lookahead Planning and Latent Learning in a Classifier System, in: J. A. Meyer, S. W. Wilson (Eds.), From Animals to Animats 1. Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90), A Bradford Book. MIT Press, 1990, pp. 316–326.
6. W. Stolzmann, An Introduction to Anticipatory Classifier Systems, in: Lanzi et al. [1], pp. 175–194.
7. W. Stolzmann, M. Butz, Latent Learning and Action-Planning in Robots with Anticipatory Classifier Systems, in: Lanzi et al. [1], pp. 303–320.
8. P. Gérard, W. Stolzmann, O. Sigaud, YACS, a new LCS using anticipation, Journal of Soft Computing Special issue on Learning Classifier Systems (to appear).
9. P. Gérard, O. Sigaud, Adding a Generalization Mechanism to YACS, San Francisco, CA, 2001, (to appear).
10. P. L. Lanzi, Learning Classifier Systems from a Reinforcement Learning Perspective , Journal of Soft Computing Special issue on Learning Classifier Systems (to appear).
11. J. H. Holmes, Evolution-Assisted Discovery of Sentinel Features in Epidemiologic Surveillance, Ph.D. thesis, Drexel Univesity (1996).
12. S. W. Wilson, Mining Oblique Data with XCS, in: Lanzi et al. [34].
13. Leigh Tesfatsion, Introductionto the CE Special Issue on Agent Based Computational Economics, Journal of Computational Economics .
14. W. B. Arthur, J. H. Holland, B. LeBaron, R. Palmer, P. Talyer, Asset Pricing Under Endogenous Expectations in an Artificial Stock Market, Tech. rep., Santa Fe Institute (1996).
15. B. Lebaron, W. B. Arthur, R. Palmer, The Time Series Properties of an Artificial Stock Market, Journal of Economic Dynamics and Control .
16. S. Schulenburg, P. Ross, Strength and Money: An LCS Approach to Increasing Returns, in: Lanzi et al. [34], extended abstract.
17. P. L. Lanzi, R. L. Riolo, A Roadmap to the Last Decade of Learning Classifier System Research (from 1989 to 1999), in: Lanzi et al. [1], pp. 33–62.
18. P. L. Lanzi, Extending the Representation of Classifier Conditions Part II: From Messy Coding to S-Expressions, in: Banzhaf et al. [35], pp. 345–352.
19. M. Dorigo, M. Colombetti, Robot Shaping: An Experiment in Behavior Engineering, MIT Press/Bradford Books, 1998.
20. S. W. Wilson, State of XCS Classifier System Research, in: Lanzi et al. [1], pp. 63–82.
21. S. W. Wilson, Classifier Systems and the Animat Problem, Machine Learning 2 (1987) 199–228.
22. T. Kovacs, Strength or Accuracy? Fitness calculation in learning classifier systems, in: Lanzi et al. [1], pp. 143–160.
23. J. H. Holmes, Learning Classifier Systems Applied to Knowledge Discovery in Clinical Research Databases, in: Lanzi et al. [1], pp. 243–264.
24. S. Saxon, A. Barry, XCS and the Monk's Problems, in: Lanzi et al. [1], pp. 223–242.
25. J. Mitlöhner, Classifier systems and economic modelling, in: Proceedings of the APL 96 conference on Designing the future, Vol. 26 (4), 1996, pp. 77–86.

26. L. Bull, On using ZCS in a Simulated Continuous Double-Auction Market, in: Banzhaf et al. [35], pp. 83–90.
27. S. Schulenburg, P. Ross, An Adaptive Agent Based Economic Model, in: Lanzi et al. [1], pp. 265–284.
28. S. W. Wilson, Generalization in the XCS classifier system, Morgan Kaufmann: San Francisco, CA, 1998, pp. 665–674.
29. M. V. Butz, T. Kovacs, P. L. Lanzi, S. W. Wilson, How XCS Evolves Accurate Classifiers, in: GECCO-2001: Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, CA, 2001.
30. T. Kovacs, M. Kerber, What makes a problem hard for XCS?, in: Lanzi et al. [34], pp. 81–101.
31. A. Tomlinson, L. Bull, A Corporate XCS, in: Lanzi et al. [1], pp. 194–208.
32. P. L. Lanzi, S. W. Wilson, Toward Optimal Classifier System Performance in Non-Markov Environments, Evolutionary Computation 8 (4) (2000) 393–418.
33. D. E. Moriarty, A. C. Schultz, J. J. Grefenstette, Evolutionary Algorithms for Reinforcement Learning, Journal of Artificial Intelligence Research 11 (1999) 199–229.
34. P. L. Lanzi, W. Stolzmann, S. W. Wilson (Eds.), Advances in Learning Classifier Systems, Vol. 1996 of LNAI, Springer-Verlag, Berlin, 2001.
35. W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, R. E. Smith (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), Morgan Kaufmann: San Francisco, CA, 1999.