Learning Classifier Systems: Looking Back and Glimpsing Ahead

Jaume Bacardit[1], Ester Bernadó-Mansilla[2], and Martin V. Butz[3]

[1] ASAP research group, School of Computer Science, Jubilee Campus, Nottingham, NG8 1BB and Multidisciplinary Centre for Integrative Biology, School of Biosciences, Sutton Bonington, LE12 5RD, University of Nottingham, UK
jaume.bacardit@nottingham.ac.uk

[2] Grup de Recerca en Sistemes Intel.ligents, Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Quatre Camins 2, 08022 Barcelona, Spain
esterb@salle.url.edu

[3] Department of Psychology, University of Würzburg, Röntgenring 11, 97070 Würzburg, Germany
butz@psychologie.uni-wuerzburg.de

Abstract. Over the recent years, research on Learning Classifier Systems (LCSs) got more and more pronounced and diverse. There have been significant advances of the LCS field on various fronts including system understanding, representations, computational models, and successful applications. In comparison to other machine learning techniques, the advantages of LCSs have become more pronounced: (1) rule-comprehensibility and thus knowledge extraction is straightforward; (2) online learning is possible; (3) local minima are avoided due to the evolutionary learning component; (4) distributed solution representations evolve; or (5) larger problem domains can be handled. After the tenth edition of the International Workshop on LCSs, more than ever before, we are looking towards an exciting future. More diverse and challenging applications, efficiency enhancements, studies of dynamical systems, and applications to cognitive control approaches appear imminent. The aim of this paper is to provide a look back at the LCS field, whereby we place our emphasis on the recent advances. Moreover, we take a glimpse ahead by discussing future challenges and opportunities for successful system applications in various domains.

1 Introduction

Learning Classifier Systems (LCSs) are robust machine learning techniques that can be applied to classification tasks [17, 6], large-scale data mining problems [80, 11], or robot control and cognitive system applications [33, 61], among others. The well-established field has its origin in John Holland's work on cognitive systems [58, 55], initiated with his seminal book on adaption in natural and artificial systems [57]. Time has seen research on several distinct approaches and paradigms. Two classic examples of these are the Michigan approach [56] versus the Pittsburgh approach [103] and also the strength-based Michigan LCSs [56] versus the more recent accuracy-based Michigan LCS [111].

Recent years have seen an explosion in quantity and diversity of LCS research. Advances have been made on various frontiers including different condition representations beyond the traditional binary/ternary rules (rules for continuous attributes [80], hyperellipsoids [28], representations based on S-expressions [78, 21], etc.), other problem classes (function approximation tasks [76, 86], clustering [109]), smarter exploration mechanisms [36, 84, 10], and various theoretical advances [34, 26, 91, 94].

The main meeting point of the LCS community, the International Workshop on Learning Classifier Systems, celebrated its 10th edition in 2007. This gives us the opportunity to take a look at the evolution of the whole LCS field from a wider perspective. In this chapter, we give an overview of the main areas of LCS research in recent years and which challenges and opportunities are laying ahead. In short, the aim of this chapter is to provide a summary of past, present, and future LCS research.

The chapter is structured as follows. Section 2 concentrates on the past by describing briefly the origins of LCS research, its motivation, development, and first successes. Section 3 surveys present LCS research. It touches on many recent advances, which we categorize along the lines of representation, learning, theory, and application. Section 4 discusses future challenges and opportunities. Based on the state-of-the-art survey, we outline various future research and application directions, which may exploit the LCS strengths and improve their weaknesses. Section 5 summarizes and concludes.

2 LCSs: Types and Approaches

John Holland, the father of LCSs, stems from the biological side and consequently introduced the LCS framework as a *cognitive systems* framework [58, 55, 56]. Inspired by principles psychology, production systems, and Darwinian evolution, he designed CSs as systems that evolve production rules in order to convert given input sensations, as well as potentially internal state representations, into useful motor commands. Rules were evaluated by basic reinforcement learning mechanisms—the infamous *bucket-brigade* algorithm [59]—and rule structure evolved by means of genetic alterations and fitness-based selection.

Due to the availability of a recent excellent LCS survey [72], rather than focusing on a historic overview on LCS research, this section gives a short introduction to the basic LCS architecture and the fundamental differences between Pittsburgh and Michigan-style LCSs. It is hoped that this section (1) forms the basis for the rest of this chapter and (2) gives a general introduction to what LCSs are.

2.1 Basic LCS Components

It might be debatable which systems may be considered LCSs. However, in order to get a grasp onto the system functionality, it seems important to identify the minimal components that are usually part of an LCS:

– A set of classifiers, that is, a set of rule-like structures, where rules usually have a condition-prediction form. This set, as it will be seen later when we describe the two main LCS paradigms, is often identified as a *population*, where each classifier in the set has its own individual identity, while other times classifiers are just part of a whole and studying them separately does not always provide good insight. For simplicity in the next paragraphs we will talk about population, even if it is not entirely appropiate.
– Classifier/population evolution mechanism, potentially enhanced with heuristics, that is designed to improve rule structures over time.
– Classifier/population evaluation mechanism, which identifies the quality of the rule or population of rules.

These components are specified in a rather general sense. However, the three components immediately imply some of the most important considerations in LCS research and application. First, the population of classifiers implies that LCSs are meant to evolve *distributed* problem solutions in which individual classifiers specify suitable subsolutions. Thus, LCS somewhat follow a mixture of experts approach. The overall solution to the problem is thus not represented in an individual rule but in the concert of rules represented in a classifier population.

Second, since rule structures are evolved by evolutionary-inspired, distributed learning techniques and this evolutionary process depends on fitness estimates, which are derived by the employed evaluation mechanism, LCSs are highly *interactive learning mechanisms*. Thus, the interaction—or the race—between sufficiently accurate evaluations and sufficiently focused evolution needs to be balanced to ensure successful learning. The various LCS systems accomplish this in some way or the other, as will be seen below.

Finally, due to the evolutionary-based structural search, LCSs usually work exceptionally competitive in problems in which either the signal for rule structures cannot be determined directly from the feedback signal, or, if a suitable feedback signal is available, directed error-based structural learners tend to get stuck in local minima. Thus, due to the interactive evaluation-evolution approach, LCSs do process feedback signals but they do not convert this signal directly into structural search biases, but use evolutionary mechanisms to induce a more thorough search that is only indirectly dependent on the feedback. Thus, LCSs are more likely to find globally-optimal solutions in particularly challenging problems, which require distributed problem solutions but in which heuristic search mechanisms tend to prematurely converge to local optima.

2.2 Michigan vs. Pittsburgh LCSs

One of the most fundamental distinction in LCS research is that of Michigan-style vs. Pittsburgh-style LCSs. Holland proposed the Michigan-style ones [56], while Kenneth DeJong and his student [103, 104] proposed the Pittsburgh-style LCS approach. Several main distinctions between the two approaches can be drawn:

- Individuals structure
- Problem solution structure
- Individuals competition/cooperation
- Online vs. offline learning

The first most fundamental distinction is the structure of an individual. While in Michigan systems each individual is a classifier, in Pittsburgh systems each individual is a set of classifiers. The other distinctions are a consequence of the first one: in Pittsburgh systems the solution to the problem is the best individual of the population. The individuals in the population compete to solve the problem and for reproductive opportinities, while in Michigan systems the solution *is* the population, that is, the classifiers in the population cooperate to solve the problem, while they compete for reproductive opportunities. The last distinction is just a consequence of the previous distinctions, and will be discussed later in this section. Figure 1 illustrates the main difference between the two systems.

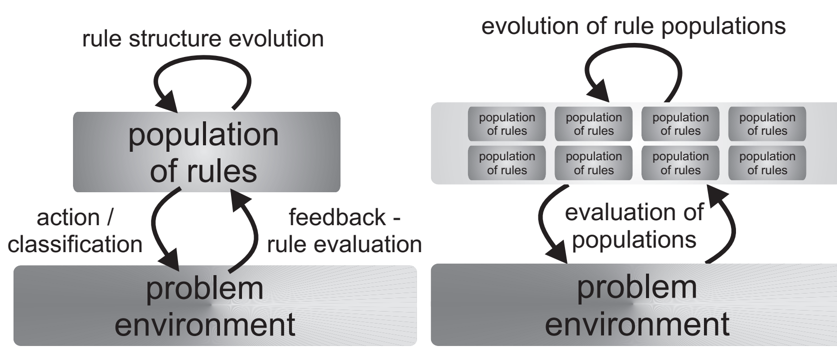

Fig. 1. While Michigan-style LCSs evolve one population of rules, in which the rules compete for offspring generation, Pittsburgh-style LCSs evaluate and evolve multiple populations, which compete with each other for reproductions.

Due to the classifier-based competition in Michigan-style LCSs, the population is usually continuously evaluated and evolved by steady-state GA techniques. Pitt-style systems, on the other hand, require longer evaluation periods until the next generation of populations can evolve, since the fitness of the whole population rather than of individual classifiers needs to be assessed. Thus, Michigan-style systems are typically applied in interactive, online learning problems while Pitt-style systems are rather suitable for offline learning problems. Nonetheless, either system has also been applied to the other problem type.

Another consequence of the rule-competition vs. population-competition difference is the typical form of the final solution. While Michigan-style systems typically evolve highly distributed problem solutions involving a rather large number of rules (typically hundreds if not more), Pitt-style systems typically evolve more compact populations involving only few rules in a population (less than a hundred). As a consequence, it can be expected that Pitt-style systems are more suitable when compact solutions with few rules are expected to solve the problem at hand. Michigan-style systems, on the other hand, are more suited if further distributed solutions are searched for.

3 Recent Advances in LCSs

This section contains an overview of the recent research in the LCS field. Our aim is to provide a spotlight of the different directions towards which the field is advancing. Thus, although our intention is to provide a good description of the overall advances of the field, for a more detailed survey including further historic remarks, the interested reader is referred to the mentioned LCS survey [72].

Classical Michigan/Pittsburgh LCS systems were rule systems with ternary condition structures, discrete actions, and real-valued predictions that used some form of evolutionary component to learn. *Present* LCS research has thoroughly analyzed these representations and mechanisms in several, often facet-wise, theoretical models. Moreover, it has gone beyond these these simple representations and is currently investigating the usage of advanced evaluation and evolution mechanisms, advanced representations, and the application to more diverse, real-world problem domains.

This section shows that the current LCS research is very diverse, tackling many different—albeit partially converging—frontiers towards which this field is advancing. We organize this section in ten subsections in which these advances can be placed. Starting from the representation of conditions, actions, and predictions, we move on to classifier competition, the evolutionary component, and theoretic considerations. We finish this section with issues on solution interpretability, efficiency enhancement techniques, and finally, move on to LCS application domains and cognitive system approaches.

3.1 Condition Structure

In this category we place the advances in the condition part of the knowledge representations. That is, the way in which the feature space is partitioned when a problem is solved. Traditionally, LCSs have used knowledge representations based on on rules for binary/nominal attributes. The ternary representation of Michigan LCSs [49, 111] or the representation of the GABIL Pittsburgh LCSs [40] are two examples of classic knowledge representations.

Over time, other kinds of knowledge representations were proposed. The main bulk of them were intended to deal with continuous attributes, something that previous representations could not do. The earliest approach for continuous attributes [115] was still a rule representation, but this time using hyperrectangles as the conditions for the classifiers. This approach has been the most popular one in recent years [106, 3, 82, 29, 14]. Other alternatives are using rule representations based on fuzzy logic [39], decision trees and synthetic instances used as the core of a nearest neighbor classifier [81], or hyperellipsoid conditions [28, 35].

Another kind of representation advance is the use of symbolic expressions to define classifier conditions [77, 2, 78, 21]. This kind of representation may be the most flexible one, in the sense that it can specify the most diverse types of problem subspaces. However, due this high diversity, it can also be considered one of the hardest to learn suitable condition structures reliably.

3.2 Action Structures

While conditions partition the problem space, actions, or classifications, propose a solution to the specified problem subspace. We analyze here the different alternatives regarding the action part of the classifiers. Traditionally, the responses of LCS systems were static. That is, in a classification problem the possible responses were the different classes of the domain. Each classifier had a static associated class. For multi-step domains, the different responses were the different discrete movements an agent could execute.

A more recent approach goes beyond these discrete action forms by proposing computed actions [73]. In this case, each classifier does not have an associated class label, but a prediction function that computes an action based on the input pattern that matched the classifier. This prediction function can be a linear combination of the inputs with a weight vector. Thus, in LCSs with computed actions, action choice does not only depend on the subspace in which a classifier condition is satisfied, but also on the action computation executed within the specified subspace.

3.3 Prediction Structure

While original LCS rules had a constant prediction, which was updated by gradient-based techniques, different kinds of advanced prediction structures and prediction estimation techniques have been employed recently. First of all, LCSs can be applied to classes of problems beyond classification/multi-step domains. The most prominent of these application domains is that of function approximation/regression tasks [116, 28, 35, 76, 86] or clustering [109]. Initial function-approximation LCS approaches tackled the regression problem as a piece-wise linear approximation, where the problem was solved by the cooperation of multiple classifiers, each of which handled a different piece of the feature space, and the continuous output of each classifier was computed as a linear combination of the input and the weight vector of the classifier. Initially, this weight vector was adjusted using a simple delta rule [116, 28, 35], although recently more sophisticated methods such as Recursive Least Squares or Kalman Filters [76] have been employed. More recently, LCSs have gone beyond linear approximations by also exploring the possibility of using polynomial predictions [74], neural predictions [87], and Support Vector Regression [86].

3.4 Classifier Competition

Given a specific input problem instance, individual rules usually propose one action and prediction. However, since usually many classifier match a certain input, another concern is the the selection of the actual action and prediction amongst all the machine classifiers available. That is, given a set of classifiers that match an input pattern, the LCS should choose the classifier/s that produce the response. In the Pittsburgh approach, the traditional solution is to organize the classifiers in a decision list [100] (an ordered set of rules), and the first classifier in the list that matches an input pattern is the one used to predict its output. In the Michigan approach, the prediction is usually made cooperatively by all the classifiers of the match or action set. Some recent advances in this topic are the usage of explicit default rules at the end of the decision list of Pittsburgh LCSs [8] or the use of better accuracy estimates of classifiers [88, 102] and principled classifier voting [19] for Michigan LCSs. Also the usage of ensemble learning methods is worth mentioning, which integrates the collective prediction of a set of models (populations of classifiers in a Michigan LCS [68] or sets of rules extracted from multiple runs of a Pittsburgh LCS [9]) using some principled fashion.

3.5 Rule Structure Evolution Mechanisms

The evolutionary mechanisms explores the space of classifier structures. In simple LCSs, this has been done by simple mutation techniques (random changes in the ternary condition representation) and simple crossover techniques (typically applying two-point crossover). Some of the recent advances, however, noted that such a simple crossover application may be disruptive, consequently applying Estimation of Distribution Algorithms (EDAs) [79], which generate a model of the problem structure and then explore the search space based on this model. There are studies of the application of EDAs for both the Michigan [36] and Pittsburgh [84] approaches. An alternative to EDAs in the context of smarter exploration mechanisms is the integration of local search techniques within an evolutionary algorithm, generally known as Memetic Algorithms [67], with examples for both Michigan [119] and Pittsburgh [10] LCSs. Also, mutation rates have been adjusted using self-adaptive mutation [22, 25]. In this case, the search operators do not improve themselves but rather are evolved to become more efficient for the current exploration mechanism.

3.6 Theory and robustness

There have been various theoretical advancements in LCSs, which gives more detailed explanations of how, why, and when an LCS works. The theoretical advancements may be separated in the analysis of the evolutionary component of the LCS system and the evaluation component.

For the evaluation component, Wilson [114, 111] has shown that his ZCS and XCS systems essentially approximate the Q-value function. Drugowitsch and Barry [42] provide an excellent mathematical foundation of the rule evaluation mechanisms in LCSs and particularly their relation to standard machine learning and adaptive filtering techniques, including Kalman filtering. Generally, rule evaluation can be considered a gradient-based, steepest-decent approximation that should adapt the prediction estimation value of a classifier maximally efficiently. Given good approximations, rule evolution can be applied successfully.

On the rule evolution side, the seminal paper on *a Theory of Generalization and Learning in XCS* [34] has shown how the evolutionary component in LCSs picks-up signals of more suitable classifier structures and consequently evolves those. Due to the strong importance of proper selection pressures, various methods have been investigated, including proportionate selection with different scaling factors [65] as well as tournament selection methods [37]. Selection pressure was explicitly modeled in [95], where tournament selection was found to be more robust than roulette wheel selection.

Meanwhile, generalization applies due to a preference of reproducing more general classifiers. Moreover, this paper has shown that a general basic support of structure needs to be available to ensure successful classifier evolution. [30] has further derived a minimal bound for the population size necessary to evolve boundedly complex classifier structures. Finally, [32] has derived another minimal population size bound that is necessary to ensure complete solution sustenance. All these bounds were used to confirm the PAC-learning capabilities of the XCS classifier system in k-DNF binary problem domains [26]. These theoretical advancements still await their extension into the real-valued realm, in which a volume-based classifier condition representation may lead to similar results.

Moreover, it has been shown that class imbalances pose some difficulties to LCS learners. Generally, learners are usually biased toward the majority class when they are exposed to domains with high class imbalances. LCSs also suffer from these difficulties, with the additional complexity of forgetting infrequent patterns caused by the incremental learning. The conditions necessary for successful learning under such conditions have been theoretically investigated in [91, 94]. In these studies, the conditions for the discovery and maintenance of minority-class niches are identified. Also a number of resampling approaches have been experimentally investigated to favor the discovery of infrequent patterns [90, 92].

3.7 Interpretability and compaction

While many efforts have relied on improving accuracy of LCSs, interpretability has also been identified as a relevant issue to get enhanced applicability of LCSs. This is an issue that have bothered both Michigan and Pittsburgh researchers. However, the approaches taken have been different. Pittsburgh LCSs usually include mechanisms for evolving compact rule sets in the search process, e.g., by means of using of minimum description length principles [7] or multiobjective approaches searching both for accurate and minimal rule sets [13, 48, 62]. On the contrary, Michigan LCSs cannot include such a direct preference for compact rule sets in the evolutionary search and thus, they usually result in large rule sets. One of the reasons is that LCSs are always performing an exploration process, so that once the evolution is stopped, the rule set contains many inexperienced rules. In such cases, Kovacs [66] suggested the use of a condensation phase, where the GA was disabled to allow for optimal rule sets. On the other hand, in domains with continuous attributes where LCSs use non-discretized representations, LCSs tend to evolve large numbers of rules that consist of many partially overlapping rules that cannot be subsumed during the exploration process. In these cases, compaction algorithms that prune excess rules with minimum loss of accuracy are proposed. The use of compaction algorithms was first proposed by Wilson for XCS with hyperrectangle representation [117] and later studied in [41, 96, 44, 120].

Fuzzy representations have been proposed as an alternative way for getting highly interpretable rule sets. There are a number of approaches using fuzzy representations in Pittsburgh and hybrid LCSs [108, 63]. In Michigan LCSs, there were early approaches such as [110, 99, 18]. Recently, fuzzy representations have been introduced in XCS [38] and later in UCS [93].

3.8 Efficiency enhancement techniques

Regarding the methods that alleviate the run-time of LCSs, many alternatives also exist. Some methods apply various kinds of windowing techniques [4] that allow the LCS to use only a subset of the training examples for fitness computation. Various policies exist to choose the training subset and the frequency in which this subset is changed. In [43] a taxonomy of such methods is given.

Parallel implementations of various LCS paradigms exist [81, 23, 80]. The GALE system [81] is an especially interesting example due to its fine-grained parallel design, where the topology and communications of the parallel model are a direct consequence of the population topology and distribution.

A widely explored efficiency enhancement approach in evolutionary computation is the use of fitness surrogates, that is, *cheap* estimators of the fitness function [64]. This approach has been recently explored within the LCS field [97, 85] by constructing fitness surrogates based on an estimated model of the problem structure. Finally, there has also been some work in speeding up the matching operations of classifier conditions for both nominal and continuous representations [83, 80] based on the usage of vectorial instructions (SSE, Altivec, etc.) available on modern day microprocessors.

3.9 Applications

With regard to applications, a clear aspect where LCSs have shown to perform competently in comparison to a broad range of machine learning techniques is in data mining tasks [17, 6]. Until recently, most of the available datasets were of relatively small size. Now, and mainly thanks to the usage of efficiency enhancement techniques explained in the previous subsection, LCSs have also been applied to much larger real datasets in bioinformatics [107] or biomedical [80] domains, containing hundreds of thousands of instances.

Other real world examples of application of LCSs are the automatic learning of fighter aircraft maneuvers [101], LCSs applied to medical domains [60] or to control problems in a steel hot strip mill [20]. There have also been some studies of the application of LCSs to stream data mining [1, 53], where there is a continuous flow of examples arriving at a very fast rate which requires that LCS learn and produce a prediction in very short time. An overview of recent applications including an extensive bibliography of LCS applications can be found elsewhere[24].

3.10 Cognitive Systems

Since Holland's introduction [58], LCSs have also played an important role in adaptive behavior research and the animat problem—research on the development of artificial animals and cognitive robots [112, 113]—and strong relations to reinforcement learning and particular online generalization in Markov decision processes have been made [70, 114, 111]. Recently, various results have shown competitive performance of XCS on benchmark machine learning problems, such as the mountain car problem [75]. Moreover, various studies have shown that XCS can maintain long reward chains and is able to generalize very well over large problem spaces [31, 29]. Thus, LCSs can be considered partially superior alternatives to standard reinforcement learning algorithms and related machine learning approaches. They have the particular advantage that the balance between GA and reward propagation and approximation can be maintained in large problem spaces, consequently learning stable payoff distributions with a highly generalized set of accurate classifiers.

There have also been advances in Partially observable Markov decision processes (PoMDP). XCS was enhanced with internal registers and has been shown to consequently evolve emergent internal representations that were able to distinguish aliasing states in the environment [71]. However, the scalability of the taken approach has not been shown and other researchers have tackled the problem with various other LCS approaches, such as a Pitt-style policy learners [69] or the AgentP classifier framework, which uses learning heuristics to overcome the PoMDP problem [121]. Despite all these efforts, the PoMDP problem is far from being solved also in the LCSs realm.

AgentP actually belongs to the class of anticipatory learning classifier systems (ALCS), which form explicit predictions about sensory consequences of actions. These systems contain classifiers that encode condition-action-next state perception triples. Various forms of ALCSs exist including the original ACS system [105], the enhanced, online generalizing ACS2 system [27], the mentioned AgentP, YCS [46], and the MACS system [45]. In comparison with policy learners, the systems have the advantage that they learn a predictive model of the environment so that they are able to flexibly adjust their behavior by simulating possible behaviors internally. This can be most effectively done with dynamic programming principles [45] but also partial updates have been investigated in accordance with the Dyna architecture in reinforcement learning [27]. For future research, it seems particularly appealing to extend these systems into real-world domains and to modularize them for to be able to efficiently represent distinct but related spaces of the environment.

4 Challenges and Opportunities

The near future points to several research challenges and various application opportunities, some of which are also shared with the machine learning community as a whole. Of common interest are issues such as applying learners beyond the traditional classification problems, extracting information from real-world datasets, system scalability, and rule selection. Besides the machine learning relation, though, advanced, modular system designs and resulting applications to complex robotics and cognitive systems tasks, amongst other domains, appear imminent. In the following, we list the, in our opinion, most promising research directions, including advanced system designs and various application opportunities. At the end of the section, we emphasize the general need in machine learning for system cookbooks, that is, principled methodologies for system applications. For LCSs in particular, the practitioner needs to be further guided to be able (1) to choose the best LCS for the problem at hand and (2) to suitably adjust the chosen LCS to optimally prepare it for the application challenge.

4.1 Problem Structure and LCS Modules

A current opportunity for LCS systems is to exploit their easy knowledge extraction possibilities to extract useful patterns for the integration of unsupervised learning and semi-supervised learning mechanisms. Some approaches have already been proposed such as those building clusters by taking advantage of the generalization capabilities of classifiers [109]. The reverse question needs to be further investigated, though, that is, if the clusters can again be used for the solution of classification problems. The XCS system essentially combines clustering and classification and clusters *for* the generation of accurate classifications.

Semi-supervised data mining approaches, where not all instances are labeled, could benefit from combined clustering plus potential classification approaches. Other frameworks, such as unsupervised learning, may also be exploited by LCSs. Mining association rules can be addressed with LCSs seeking for the most frequent patterns among the attributes. In these cases, generalization is a key issue, and LCSs are ready to conquer large problem spaces with the appropriate generalization mechanisms. Thus, LCSs are ready to be applied to domains in which partially pure clustering and partially problem space clustering for accurate classifications or predictions are necessary.

So far most LCSs have been flat, processing input and converting that input into a classification, behavior, or prediction. The great problem space structuring capabilities, however, suggest the generation of more modularized and hierarchical LCSs. That is, since LCSs have been shown to be able to automatically identify building-block structures in problem domains [36], it appears imminent that modular LCS systems make these structures explicit, abstract them, and use the abstracted concepts for further processing. A first approach in this direction can be found in this book [98]. Due to the LCS principle of clustering-for-prediction, however, more modular and hierarchical LCSs, which may process subsets of input dimensions, abstract the information, and merge it in higher level structures seem possible. Once such system architectures will be successfully designed, a whole new dimension of LCS systems and successful LCS applications will be at hand.

While such modular, hierarchical LCS systems may be applied to various problem domains, the application in the cognitive systems realm appears most appealing. Over the recent years, research in cognitive neuroscience, psychology, and the mind in general has emphasized two very important aspects of brain structure and functionality: interactive modularity and sensorimotor codes [47, 51, 52, 89, 118]. That is, the brain structures sensory and motor information in various modules, whereby the purpose of these modules is to satisfy motivational modules and serve control modules for successful behavioral executions. Thus, many sensorimotor codes are found in the brain, which encode the dependence of sensory information on motor commands in various forms and multimodal modules. LCSs can structure sensory information for successful prediction and motor control. Thus, they have the potential to directly develop sensorimotor codes. Once advanced system modularity and further interactivity of LCS systems is realized, then also the interactive modularity of sensorimotor codes may be mimicked. Thus, the design of advanced, cognitive LCSs appears to be within our grasp.

4.2 LCS Cookbook

Many different learning algorithms have been proposed and evaluated experimentally in a number of domains. No difference in LCSs: Various LCSs have been proposed and applied to various problem domains—each of which with some claimed superiorities shown by some evaluations in suitable problems or problem classes. With such a variety of available methods, the practitioner finds it difficult to choose a learner for a given application. Which LCS is better suited for a given problem? Which are the conditions of applicability of an LCS? Although we wish to be able to give exact answers to these questions, it is currently still a big challenge to give precise system design recommendations given a particular problem. In fact, often the trouble already starts at the problem definition itself and particularly the to-be expected problem structures.

Thus, a big challenge is the development of further theoretical understanding of which types of problems exist and which kind of LCS, or learning method in general, is most suited to solve each type of problem. Goldberg has approached this challenge with the definition of boundedly difficult problems in various optimization problem domains [50]. The theoretical performance analyses of the XCS classifier system have moved along a similar vein and identified various problem properties that influence problem difficulty [29]. Another approach works on the categorization of problems by means of geometrical descriptors, such as the separability of classes or the discriminant power of attributes [15, 54, 12]. These works have identified some features that are critical to the success of learners and can act as predictors of the learners' performance [16]. However, there is much more work needed to further understand the intrinsic characteristics of data and find the key properties relevant for the identification of the most potent learning algorithm.

Another dimension of problem complexity, rather simpler to describe but nonetheless equally difficult for LCS systems, is the size of the problem. That is, how can we make sure that LCS performance is not degraded when tackling problems of larger sizes. Scalability analysis of LCSs in such domains is still at its beginning but the theoretical knowledge on different facets of the problem is available [5, 29, 94]. Thus, compound approaches that tackle all the problem facets in real-world data mining applications are pending.

Further extensions of such investigations in the LCS realm appear in close reach, including further analyses of adversarial LCS problems and further studies of which features are critical for LCS success. These studies are expected to close the gap between LCSs and other machine learning methods. Moreover, they are expected to lead towards more precise knowledge of the relative strengths and weaknesses of the available learning systems and, in particular, the domains of competence of different LCS learners. On this road, we expect that LCSs and evolutionary approaches for machine learning will be progressively better known and become accepted in the machine learning community as a whole. Meanwhile, the particular strengths of LCSs will become appreciated, such as learning robustness, versatility due to the availability of several representations and balanced learning influences, and the explanatory power of LCSs.

4.3 Data Mining

Clearly the available applications in the data mining domain have not exploited the full potential of LCSs. Knowledge extraction and exploitation are still at the beginning. Moreover, the systems' versatility has not been exploited to its fullest.

In particular, it appears that there is still a lot of room for applying LCSs in real-world domains, far beyond studies that are based on the known *toy* problems from the UCI repository. Although the problems from the UCI repository have been labeled "real-world problems", and although they technically mostly are, they do not fully represent the difficulties of real-world data mining applications. LCSs may be applied to mine interesting patterns from very large datasets, containing hundreds of thousands of registers and a great number of attributes, plus all associated complexities altogether (including high instance imbalances, noise, missing information, partially labeled instances, etc).

Searching patterns through datasets structured in some manner, different from the usual plain file, can also be a difficult challenge to any learning scheme. Data may be presented in complex structures. While the traditional form consists of a number of instances, each characterized by a fixed number of attributes with associated class, data is now often presented in more intricate ways. Medical records contain diverse data sets, which were collected from many sources: some may have a variable number of medical tests associated, others may contain results from related tests performed over a variable number of individuals of the same family, etc. Thus, incomplete information, multiple-instance learning, and varied types of data are some of the difficulties that LCSs will need to face.

These are, in our opinion, the data mining-related challenges for LCS. As we have mentioned in section 3.9, there are some initial examples of LCS application to large-scale real datasets. These examples have shown that, indeed, LCS can be applied successfully to these kind of domains, providing accurate solutions and high explanatory power due to its rule-based representations. However, one question remains unanwered in a broad and systematic sense: how can we guarantee that our LCS are well adjusted when applied to data mining domains? The answer is not simple, but we think that the challenges that we presented in the first to subsections of this section are important steps towards this answer, specially the cookbook part. If we are able to (1) determine which LCS modules/paradigms are more suited for the domain at hand thanks to all the problem complexity metrics and (2) know how to parametrize our LCSs appropiately by using principled policies derived from the theoretical analysis made from each LCS component/faced, then we will have surpassed an important milestone towards successful LCS application to data mining tasks and, even more important, we will provide the community with sound instructions of how to get the best of the available LCS technology, which can help to broaden its acceptance and use in the scientific community.

5 Conclusions

LCSs have come a long way. While the first LCSs were mainly biologically inspired and designed as admittedly simple but flexible adaptive systems, modern LCS applications focused mainly on the datamining challenge. Over the last decade or so, LCS research has progressed towards a solid system understanding, it has created a theoretical foundation of LCS learning concepts, and it has shown LCS competitiveness in various machine learning challenges.

While there are still challenges to be solved, we believe that these challenges are actual opportunities for future successful research efforts and even potentially groundbreaking system applications. LCSs are ready to solve complex real-world problems in the datamining domain but also in the cognitive systems domain and others. The rest of this book provides a great overview of current research advances and application approaches. Various pointers to further recent literature are available throughout the book. Thus, we hope that these IWLCS post-workshop proceedings once again give a useful overview of current system progresses and encourage further effort along the plotted research directions. Only future research can ultimately verify the apparent opportunities ahead.

Acknowledgments J.Bacardit acknowledges the support of the UK Engineering and Physical Sciences Research Council (EPSRC) under grant GR/T07534/01. E.Bernad-Mansilla acknowledges the support of *Enginyeria i Arquitectura La Salle, Universitat Ramon Llull*, and the *Ministerio de Educación y Ciencia* under grant TIN2005-08386-C05-04. M.Butz acknowledges funding from the Emmy Noether program of the German research foundation (grant BU1335/3-1) and likes to thank his colleagues at the department of psychology and the COBOSLAB team.

References

1. Abbass, H.A., Bacardit, J., Butz, M.V., Llora, X.: Online adaption in learning classifier systems: Stream data mining. Technical Report 2004031, Illinois Genetic Algorithms Lab, University of Illinois at Urbana-Champaign (2004)
2. Ahluwalia, M., Bull, L.: A genetic programming-based classifier system. In: Proceedings of the Genetic and Evolutionary Computation Conference. Volume 1., Morgan Kaufmann (1999) 11–18
3. Bacardit, J., Garrell, J.M.: Analysis and improvements of the adaptive discretization intervals knowledge representation. In: GECCO 2004: Proceedings of the Genetic and Evolutionary Computation Conference, Springer-Verlag, LNCS 3103 (2004) 726–738
4. Bacardit, J., Goldberg, D., Butz, M., Llorà, X., Garrell, J.M.: Speeding-up pittsburgh learning classifier systems: Modeling time and accuracy. In: Parallel Problem Solving from Nature - PPSN 2004, Springer-Verlag, LNCS 3242 (2004) 1021–1031
5. Bacardit, J.: Pittsburgh Genetics-Based Machine Learning in the Data Mining era: Representations, generalization, and run-time. PhD thesis, Ramon Llull University, Barcelona, Catalonia, Spain (2004)
6. Bacardit, J., Butz, M.V.: Data mining in learning classifier systems: Comparing xcs with gassist. In: Advances at the frontier of Learning Classifier Systems. Springer-Verlag (2007) 282–290
7. Bacardit, J., Garrell, J.M.: Bloat control and generalization pressure using the minimum description length principle for a pittsburgh approach learning classifier system. In: Proceedings of the 6th International Workshop on Learning Classifier Systems, (in press), LNAI, Springer-Verlag (2003)
8. Bacardit, J., Goldberg, D.E., Butz, M.V.: Improving the performance of a pittsburgh learning classifier system using a default rule. In: Learning Classifier Systems, Revised Selected Papers of the International Workshop on Learning Classifier Systems 2003-2005. Springer-Verlag, LNCS 4399 (2007) 291–307

9. Bacardit, J., Krasnogor, N.: Empirical evaluation of ensemble techniques for a pittsburgh learning classifier system. In: Ninth International Workshop on Learning Classifier Systems (IWLCS 2006). Lecture Notes in Artificial Intelligenge, Springer (2006) to appear.
10. Bacardit, J., Krasnogor, N.: Smart crossover operator with multiple parents for a pittsburgh learning classifier system. In: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2006) 1441–1448
11. Bacardit, J., Stout, M., Krasnogor, N., Hirst, J.D., Blazewicz, J.: Coordination number prediction using learning classifier systems: performance and interpretability. In: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM Press (2006) 247–254
12. Basu, M., Ho, T.K.E.: Data Complexity in Pattern Recognition. Springer (2006)
13. Bernadó-Mansilla, E., Llorà, X., Traus, I.: Multiobjective Learning Classifier Systems. In: Multi-Objective Machine Learning. Volume 16 of Studies in Computational Intelligence., Springer (2006) 261–288
14. Bernadó-Mansilla, E., Garrell-Guiu, J.M.: Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks. Evolutionary Computation **11** (2003) 209–238
15. Bernadó-Mansilla, E., Ho, T.K.: Domain of Competence of XCS Classifier System in Complexity Measurement Space. IEEE Transactions on Evolutionary Computation **9** (2005) 82–104
16. Bernadó-Mansilla, E., Kam Ho, T.: On Classifier Domains of Competence. In: Proceedings of the 17th International Conference on Pattern Recognition. Volume 1. (2004) 136–139
17. Bernadó-Mansilla, E., Llorà, X., Garrell, J.M.: XCS and GALE: a comparative study of two learning classifier systems with six other learning algorithms on classification tasks. In: Fourth International Workshop on Learning Classifier Systems - IWLCS-2001. (2001) 337–341
18. Bonarini, A.: Evolutionary Learning of Fuzzy rules: competition and cooperation. In: Fuzzy Modelling: Paradigms and Practice, Norwell, MA: Kluwer Academic Press (1996) 265–284
19. Brown, G., Kovacs, T., Marshall, J.A.R.: Ucspv: principled voting in ucs rule populations. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 1774–1781
20. Browne, W.: The development of an industrial learning classifier system for datamining in a steel hot strip mill. In Bull, L., ed.: Applications of Learning Classifier Systems. Springer-Verlag (2004) 223–259
21. Browne, W.N., Ioannides, C.: Investigating scaling of an abstracted lcs utilising ternary and s-expression alphabets. In: GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 2759–2764
22. Bull, L., Hurst, J., Tomlison, A.: Self-adaptive mutation in classifier system controllers. In Meyer, J.A., Berthoz, A., Floreano, D., Roitblatt, H., Wilson, S., eds.: From Animals to Animats 6 - The Sixth International Conference on the Simulation of Adaptive Behaviour, MIT Press (2000)
23. Bull, L., Studley, M., Bagnall, A., Whittley, I.: Learning classifier system ensembles with rule-sharing. Evolutionary Computation, IEEE Transactions on **11** (2007) 496–502

24. Bull, L., ed.: Applications of Learning Classifier Systems. (2004)
25. Bull, L.: On lookahead and latent learning in simple lcs. In: GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, New York, NY, USA, ACM (2007) 2633–2636
26. Butz, M.V., Goldberg, D.E., Lanzi, P.L.: Computational complexity of the XCS classifier system. In Bull, L., Kovacs, T., eds.: Foundations of Learning Classifier Systems. Studies in Fuzziness and Soft Computing. (2005) 91–126
27. Butz, M.V.: Anticipatory learning classifier systems. Kluwer Academic Publishers, Boston, MA (2002)
28. Butz, M.V.: Kernel-based, ellipsoidal conditions in the real-valued xcs classifier system. In: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2005) 1835–1842
29. Butz, M.V.: Rule-Based Evolutionary Online Learning Systems: A Principled Approach to LCS Analysis and Design. (2006)
30. Butz, M.V., Goldberg, D.E.: Bounding the population size in XCS to ensure reproductive opportunities. (2003) 1844–1856
31. Butz, M.V., Goldberg, D.E., Lanzi, P.L.: Gradient descent methods in learning classifier systems: Improving XCS performance in multistep problems. IEEE Transactions on Evolutionary Computation **9** (2005) 452– 473
32. Butz, M.V., Goldberg, D.E., Lanzi, P.L., Sastry, K.: Problem solution sustenance in XCS: Markov chain analysis of niche support distributions and the impact on computational complexity. Genetic Programming and Evolvable Machines **8** (2007) 5–37
33. Butz, M.V., Hoffmann, J.: Anticipations control behavior: Animal behavior in an anticipatory learning classifier system. Adaptive Behavior **10** (2002) 75–96
34. Butz, M.V., Kovacs, T., Lanzi, P.L., Wilson, S.W.: Toward a theory of generalization and learning in XCS. IEEE Transactions on Evolutionary Computation **8** (2004) 28–46
35. Butz, M.V., Lanzi, P.L., Wilson, S.W.: Hyper-ellipsoidal conditions in xcs: rotation, linear approximation, and solution structure. In: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2006) 1457–1464
36. Butz, M.V., Pelikan, M., Llorà, X., Goldberg, D.E.: Automated global structure extraction for effective local building block processing in xcs. Evol. Comput. **14** (2006) 345–380
37. Butz, M.V., Sastry, K., Goldberg, D.E.: Strong, stable, and reliable fitness pressure in XCS due to tournament selection. Genetic Programming and Evolvable Machines **6** (2005) 53–77
38. Casillas, J., Carse, B., Bull, L.: Fuzzy-xcs: A michigan genetic fuzzy system. IEEE Transactions on Fuzzy Systems **15** (2007) 536–550
39. Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific (2001)
40. DeJong, K.A., Spears, W.M.: Learning concept classification rules using genetic algorithms. In: Proceedings of the International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1991) 651–656
41. Dixon, P.W., Corne, D.W., Oates, M.J.: A Preliminary Investigation of Modified XCS as a Generic Data Mining Tool. In Lanzi, P., Stolzmann, W., Wilson, S., eds.: Advances in Learning Classifier Systems, 4th International Workshop. Volume 2321 of Lecture Notes in Computer Science., Springer (2002) 133–150

42. Drugowitsch, J., Barry, A.: A formal framework and extensions for function approximation in learning classifier systems. Machine Learning **70** (2008) 45–88
43. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag (2002)
44. Fu, C., David, L.: A Modified Classifier System Compaction Algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers Inc. (2002) 920–925
45. Gérard, P., Meyer, J.A., Sigaud, O.: Combining latent learning and dynamic programming in MACS. European Journal of Operational Research **160** (2005) 614–637
46. Gérard, P., Sigaud, O.: Adding a generalization mechanism to YACS. (2001) 951–957
47. Ghahramani, Z., Wolpert, D.M.: Modular decomposition in visuomotor learning. Nature (1997) 392–395
48. Ghosh, A., Nath, B.: Multi-objective rule mining using genetic algorithms. Information Sciences **163** (2004) 123–133
49. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc. (1989)
50. Goldberg, D.E.: The Design of Innovation: Lessons from and for Competent Genetic Algorithms. Kluwer Academic Publishers (2002)
51. Grush, R.: The emulation theory of representation: Motor control, imagery, and perception. Behavioral and Brain Sciences **27** (2004) 377–96
52. Haruno, M., Wolpert, D.M., Kawato, M.: Hierarchical mosaic for movement generation. In Ono, T., Matsumoto, G., Llinas, R., Berthoz, A., Norgren, R., Nishijo, H., Tamura, R., eds.: Excepta Medica International Coungress Series. Volume 1250., Amsterdam, The Netherlands, Elsevier Science B.V. (2003) 575–590
53. H.Dam, H., Lokan, C., Abbas, H.A.: Evolutionary online data mining: An investigation in a dynamic environment. In: Evolutionary Computation in Dynamic and Uncertain Environments. Springer-Verlag (2007) 153–178
54. Ho, T.K., Basu, M.: Measuring the complexity of classification problems. In: 15th International Conference on Pattern Recognition. (2000) 43–47
55. Holland, J.H.: A cognitive system with powers of generalization and adaptation. Unpublished manuscript (1977)
56. Holland, J.H., Reitman, J.S.: Cognitive systems based on adaptive algorithms. In Hayes-Roth, D., Waterman, F., eds.: Pattern-directed Inference Systems. Academic Press, New York (1978) 313–329
57. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
58. Holland, J.H.: Adaptation. In Rosen, R., Snell, F., eds.: Progress in theoretical biology. Volume 4. Academic Press, New York (1976) 263–293
59. Holland, J.H.: Properties of the bucket brigade algorithm. (1985) 1–7
60. Holmes, J.H., Durbin, D.R., Winston, F.K.: The learning classifier system: an evolutionary computation approach to knowledge discovery in epidemiologic surveillance. Artificial Intelligence In Medicine **19** (2000) 53–74
61. Hurst, J., Bull, L.: A neural learning classifier system with self-adaptive constructivism for mobile robot learning. Artificial Life **12** (2006) 1–28
62. Ishibuchi, H., Nakashima, T., Murata, T.: Three-objective genetics-based machine learning for linguistic rule extraction. Information Sciences **136** (2001) 109–133
63. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. International Journal of Approximate Reasoning **44** (2007) 4–31

64. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput. **9** (2005) 3–12
65. Kharbat, F., Bull, L., Odeh, M.: Revisiting genetic selection in the xcs learning classifier system. Proceedings of the IEEE Congress on Evolutionary Computation **CEC 2005** (2005) 2061–2068
66. Kovacs, T.: XCS Classifier System Reliably Evolves Accurate, Complete and Minimal Representations for Boolean Functions. In Roy, R., Chawdhry, P., Pant, R., eds.: Soft Computing in Engineering Design and Manufacturing, Springer-Verlag (1997) 59–68
67. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy and design issues. IEEE Transactions on Evolutionary Computation **9** (2005) 474– 488
68. L. Bull, M. Studley, A.J.B., Whittley, I.: On the use of rule sharing in learning classifier system ensembles. In: Proceedings of the 2005 Congress on Evolutionary Computation. (2005)
69. Landau, S., Picault, S., Sigaud, O., Gérard, P.: Further comparison between ATNoSFERES and XCSM. (2003) 99–117
70. Lanzi, P.L.: An analysis of generalization in the XCS classifier system. Evolutionary Computation **7** (1999) 125–149
71. Lanzi, P.L.: Adaptive agents with reinforcement learning and internal memory. (2000) 333–342
72. Lanzi, P.L.: Learning classifier systems: then and now. Evolutionary Intelligence **1** (2008) 63,82
73. Lanzi, P.L., Loiacono, D.: Classifier systems that compute action mappings. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 1822–1829
74. Lanzi, P.L., Loiacono, D., Wilson, S.W., Goldberg, D.E.: Extending xcsf beyond linear approximation. In: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, New York, NY, USA, ACM (2005) 1827–1834
75. Lanzi, P.L., Loiacono, D., Wilson, S.W., Goldberg, D.E.: Classifier prediction based on tile coding. (2006) 1497–1504
76. Lanzi, P.L., Loiacono, D., Wilson, S.W., Goldberg, D.E.: Prediction update algorithms for XCSF: RLS, kalman filter, and gain adaptation. In: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2006) 1505–1512
77. Lanzi, P.L., Perrucci, A.: Extending the representation of classifier conditions part ii: From messy coding to s-expressions. In Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E., eds.: Proceedings of the Genetic and Evolutionary Computation Conference. Volume 1., Orlando, Florida, USA, Morgan Kaufmann (1999) 345–352
78. Lanzi, P.L., Rocca, S., Solari, S.: An approach to analyze the evolution of symbolic conditions in learning classifier systems. In: GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 2795–2800
79. Larranaga, P., Lozano, J., eds.: Estimation of Distribution Algorithms, A New Tool for Evolutionnary Computation. Genetic Algorithms and Evolutionnary Computation. Kluwer Academic Publishers (2002)
80. Llorà, X., Priya, A., Bhargava, R.: Observer-invariant histopathology using genetics-based machine learning. Natural Computing, Special issue on Learning Classifier Systems (2008) in press

81. Llorà, X., Garrell, J.M.: Knowledge-independent data mining with fine-grained parallel evolutionary algorithms. In: Proceedings of the Third Genetic and Evolutionary Computation Conference, Morgan Kaufmann (2001) 461–468
82. Llorà, X., Reddy, R., Matesic, B., Bhargava, R.: Towards better than human capability in diagnosing prostate cancer using infrared spectroscopic imaging. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 2098–2105
83. Llorà, X., Sastry, K.: Fast rule matching for learning classifier systems via vector instructions. In: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2006) 1513–1520
84. Llorà, X., Sastry, K., Goldberg, D.E., delaOssa, L.: The x-ary extended compact classifier system: Linkage learning in pittsburgh lcs. In: Proceedings of the 9th International Workshop on Learning Classifier Systems - IWLCS2006, (in press), LNAI, Springer-Verlag (2006)
85. Llorà, X., Sastry, K., Yu, T.L., Goldberg, D.E.: Do not match, inherit: fitness surrogates for genetics-based machine learning techniques. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM (2007) 1798–1805
86. Loiacono, D., Marelli, A., Lanzi, P.L.: Support vector regression for classifier prediction. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 1806–1813
87. Luca Lanzi, P., Loiacono, D.: Xcsf with neural prediction. Evolutionary Computation, 2006. CEC 2006. IEEE Congress on (0-0 0) 2270–2276
88. Marshall, J.A.R., Brown, G., Kovacs, T.: Bayesian estimation of rule accuracy in ucs. In: GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 2831–2834
89. O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behavioral and Brain Sciences **24** (2001) 939–1031
90. Orriols-Puig, A., Bernadó-Mansilla, E.: The Class Imbalance Problem in UCS Classifier System: Fitness Adaptation. In: Proceedings of the 2005 Congress on Evolutionary Computation. Volume 1., IEEE (2005) 604–611
91. Orriols-Puig, A., Bernadó-Mansilla, E.: Bounding XCS's Parameters for Unbalanced Datasets. In: Proceedings of the 2006 Genetic and Evolutionary Computation Conference. Volume 2., ACM Press (2006) 1561–1568
92. Orriols-Puig, A., Bernadó-Mansilla, E.: The Class Imbalance Problem in Learning Classifier Systems: A Preliminary Study. In: Advances at the frontier of LCS. Volume 4399 of Lecture Notes in Computer Science., Springer (2006) 164–183
93. Orriols-Puig, A., Casillas, J., Bernadó-Mansilla, E.: Fuzzy-UCS: A Michigan-style Learning Fuzzy-Classifier System for Supervised Learning. IEEE Transactions on Evolutionary Computation (2008) in press
94. Orriols-Puig, A., Goldberg, D., Sastry, K., Bernadó-Mansilla, E.: Modeling XCS in Class Imbalances: Population Size and Parameter Settings. In: Proceedings of the 2007 Genetic and Evolutionary Computation Conference. Volume 2., ACM Press (2007) 1838–1845
95. Orriols-Puig, A., Sastry, K., Lanzi, P., Goldberg, D., Bernadó-Mansilla, E.: Modeling selection pressure in xcs for proportionate and tournament selection. In: Proceedings of the 2007 Genetic and Evolutionary Computation Conference. Volume 2., ACM Press (2007) 1846–1853

96. Orriols Puig, A., Bernadó-Mansilla, E.: Analysis of Reduction Algorithms in XCS Classifier System. In: Recent Advances in Artificial Intelligence Research and Development. Volume 113 of Frontiers in Artificial Intelligence and Applications., IOS Press (2004) 383–390
97. Orriols-Puig, A., Bernadó-Mansilla, E., Sastry, K., Goldberg, D.E.: Substructrual surrogates for learning decomposable classification problems: implementation and first results. In: GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, New York, NY, USA, ACM (2007) 2875–2882
98. Orriols-Puig, A., Sastry, K., Goldberg, D.E., Bernadó-Mansilla, E.: Substructural surrogates for learning decomposable classification problems. In Bacardit, J., Bernad'o-Mansilla, E., Butz, M.V., eds.: this volume. (2008)
99. Parodi, A., Bonelli, P.: A new approach to fuzzy classifier systems. In: 5th International Conference on Genetic Algorithms, Morgan Kaufmann (1993) 223–230
100. Rivest, R.L.: Learning decision lists. Machine Learning **2** (1987) 229–246
101. Smith, R.E., El-Fallah, A., Ravichandran, B., Mehra, R., Dike, B.A.: The fighter aircraft lcs: A real-world, machine innovation application. In Bull, L., ed.: Applications of Learning Classifier Systems. Springer-Verlag (2004) 113–142
102. Smith, R.E., Jiang, M.K.: A learning classifier system with mutual-information-based fitness. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (25-28 Sept. 2007) 2173–2180
103. Smith, S.F.: A Learning System Based on Genetic Algorithms. PhD thesis, University of Pittsburgh (1980)
104. Smith, S.F.: Flexible learning of problem solving heuristics through adaptive search. In: Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Los Altos, CA, Morgan Kaufmann (1983) 421–425
105. Stolzmann, W.: Anticipatory classifier systems. Genetic Programming 1998: Proceedings of the Third Annual Conference (1998) 658–664
106. Stone, C., Bull, L.: For real! XCS with continuous-valued inputs. Evolutionary Computation Journal **11** (2003) 298–336
107. Stout, M., Bacardit, J., Hirst, J.D., Krasnogor, N.: Prediction of recursive convex hull class assignments for protein residues. Bioinformatics **In press** (2008)
108. Suzuki, T., Kodama, T., Furuhashi, T., Tsut, H.: Fuzzy modeling using genetic algorithms with fuzzy entropy as conciseness measure. Information Sciences **136** (2001) 53–67
109. Tamee, K., Bull, L., Pinngern, O.: Towards clustering with xcs. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA, ACM Press (2007) 1854–1860
110. Valenzuela-Rendón, M.: The fuzzy classifier system: A classifier system for continuously varying variables. In: Fourth International Conference on Genetic Algorithms (ICGA), Morgan Kaufmann (1991) 346–353
111. Wilson, S.W.: Classifier fitness based on accuracy. Evolutionary Computation **3** (1995) 149–175
112. Wilson, S.W.: Knowledge growth in an artificial animal. Proceedings of an International Conference on Genetic Algorithms and Their Applications (1985) 16–23
113. Wilson, S.W.: Classifier systems and the animat problem. Machine Learning **2** (1987) 199–228
114. Wilson, S.W.: ZCS: A zeroth level classifier system. Evolutionary Computation **2** (1994) 1–18

115. Wilson, S.W.: Get real! XCS with continuous-valued inputs. In Booker, L., Forrest, S., Mitchell, M., Riolo, R.L., eds.: Festschrift in Honor of John H. Holland, Center for the Study of Complex Systems (1999) 111–121
116. Wilson, S.W.: Classifiers that approximate functions. Natural Computing: an international journal **1** (2002) 211–234
117. Wilson, S.W.: Compact Rulesets from XCSI. In Lanzi, P., Stolzmann, W., Wilson, S., eds.: Advances in Learning Classifier Systems, 4th International Workshop. Volume 2321 of Lecture Notes in Artificial Intelligence., Springer (2002) 197–210
118. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks **11** (1998) 1317–1329
119. Wyatt, D., Bull, L.: A memetic learning classifier system for describing continuous-valued problem spaces. In Hart, W., Krasnogor, N., Smith, J., eds.: Recent Advances in Memetic Algorithms. Springer (2004) 355–396
120. Wyatt, D., Bull, L., Parmee, I.: Building Compact Rulesets for Describing Continuous-Valued Problem Spaces Using a Learning Classifier System. In Parmee, I., ed.: Adaptive Computing in Design and Manufacture. Volume VI., Springer (2004) 235–248
121. Zatuchna, Z.V.: AgentP: A Learning Classifier System with Associative Perception in Maze Environments. PhD thesis, School of Computing Sciences, UEA (2005)