Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2003 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2003

Assessing the Cognitive Abilities of Alternate Learning Classifier System Architectures

Dave Gaines
University of Kentucky

ASSESSING THE COGNITIVE ABILITIES OF ALTERNATE LEARNING CLASSIFIER SYSTEM ARCHITECTURES

Dave Gaines
University of Kentucky
dgaines@uky.edu

Background

Since its inception in the 1960s, the Genetic Algorithm (GA) framework for solving complex problems has been simultaneously intensely studied and deployed. Despite wide-ranging practical successes in engineering, manufacturing, applied, and social science domains, developing GA-based systems has been more art than science. Consequently, some researchers have attempted to build and test theories and models for "robust" GA design.

Given this attention to "pure" Genetic Algorithm research and implementations, progress on a subsequent GA-based framework called Learning Classifier Systems (LCS) lay dormant until the late 1990s. Stalwarts in GA/LCS research have opined that to further advance the field and facilitate theory formation, a broad study of LCSs, particularly one that focuses on their cognitive aspects, is needed.

I wish to contribute to this theory-building effort by examining, using simulation modeling and analyses, how alternative LCS architectures learn to cope with other artificial entities in challenging, artificial environments created using variants of the Iterated Prisoner's Dilemma (IPD) Tournament setting.. The use of competing entities in this setting may be likened to a number of practical applications in which different agents must negotiate or compete with each other. One possible application is the use of computer-based agents in negotiations in a buying-selling situation. In such an environment, a buyer's agent must attempt to discern the seller's negotiation pattern, and then use this information to accomplish its objective. In this example, an LCS-based agent could be used in repeated encounters with the seller to improve its performance with regard to a measure of interest such as price, quantity, or delivery time.

Section 1: The Traditional Learning Classifier System

The original Learning Classifier System (LCS) model dates back to the early 1970s and is well documented; a complete description appears in Booker, Goldberg, and Holland (Booker, Goldberg et al. 1989). A brief description of LCS is provided here.

An LCS is a special form of rule-based system which can self-evolve its rule set over time based on interactions with its operating environment. Each rule, expressed in conventional IF-THEN form, is called a classifier and is usually cast using a limited, ternary vocabulary set—1, 0, and #—where combinations of 1s and 0s convey meaning and the # symbol denotes "unknown" or "irrelevant."

An LCS system interacts with its operating environment via Detectors and Effectors. A Detector enables the LCS to process relevant environmental stimuli, cast the stimuli in binary (i.e., 1, 0) language, and post a fixed length message to a Message List. An Effector is triggered by appropriate messages in the list to take some action ("a"), which may or may not result in earning some reward (potentially negative) from the environment. The Rule Base holds a population of classifiers. Associated with each classifier is a measure of its utility (i.e., strength or fitness, S). These fitness values are adjusted over time to reflect each classifier's relative utility in the society of classifiers. Classifiers in the Rule Base cooperate in one sense (e.g., as enablers and reward sharers) but also compete in another (pay taxes and bid against one another). While all classifiers in the starting Rule Base usually have the same strength, strength values can be adjusted upward, downward, or remain stationary over time. At a suitable point (e.g., when strength values have stabilized, when a certain number of firings have occurred, etc.), the LCS's activity is

stopped and its Rule Base replenished. This is usually done using a Genetic Algorithm (GA). The ultimate goal of an LCS system is to evolve a set of classifiers which "solve" a particular problem.

Section 2: Perceived Drawbacks of the Traditional LCS Paradigm

As a consequence of recent LCS research, several supposed weaknesses of the original LCS model have been identified (Jantke and Lange 1995; Wilson 1995). These potential drawbacks relate to the traditional practice of associating a classifier's strength as a measure of its utility and allowing higher strength classifiers relatively greater opportunity to fire as well as to engage in genetic procreation. Because strength is directly related to payoff magnitude, such LCSs may be characterized as payoff-magnitude driven LCSs. The criticisms of payoff-magnitude driven LCSs are summarized below:

(a) It is possible that the environment contains niches (a set of states at each of which a common subset of available classifiers are able to match and are, therefore, all candidates for firing). Some niches could offer greater payoffs to the LCS than others. It is possible for classifiers operating in such niches to dominate the population during genetic procreation as they gain higher rewards and grow disproportionately fitter over time. Lower strength rules, upon which overall system performance could depend, are purged.

(b) One way of mitigating the drawback in (a) is to share the portion of any accrued reward intended solely for the firing classifier with all classifiers in the Match Set that advocate the same action as the firing classifier. The hope is that since the payoff is divided between multiple, "equivalent" classifiers, no single classifier would grow dominant. However, this solution introduces another weaknes—a single classifier's strength now becomes a weaker (indirect) measure of its utility and can no longer be used as a surrogate for its payoff-generation ability (i.e., as a predictor of its utility); this ability is now distributed amongst several classifiers. We must now find a measure other than strength to assess a classifier's utility.

(c) In situations were rule-chaining is essential (i.e., deferred payoff systems), early enabling rules in a lengthy chain will appear less fit over time, even with a reward back-propagation mechanism in place that offers some of the reward to the enablers. Thus, when the GA module is invoked, its parent selection mechanism tends to ignore the relatively weaker enablers despite the fact that they are crucial to system success. Useful genetic material is often lost as a result. A solution to this problem is to use the GA on Match Sets rather than on the entire population of classifiers. Thus, there will be no procreation-related competition between Match Sets where classifiers in one set dominate those in the others. Such use of a GA is termed as "niche GAs." Even with niching, two problems remain.

(d) The GA component of the LCS is unable to distinguish between more specific classifiers having a certain payoff accuracy from more general versions (i.e., having more # symbols in their conditions) that offer the same payoff, on average. Consequently, because the more general versions tend to match environmental states more often than the more specific ones, the more general versions tend to proliferate over time.

(e) Generalizations appear to be desirable. However, there is no mechanism to assure that the generalizations are good performers in the sense that their actions yield payoffs close to what is expected when they are fired. That is, with a payoff-magnitude driven LCS, there is no mechanism in place to ensure that accurate generalizations are evolved.

Section 3: The XCS Paradigm

In response to these concerns and for other reasons, researchers have developed alternative LCS models. Among the more noteworthy attempts are latent-learning LCSs by Riolo (Riolo 1991), payoff-accuracy driven LCSs by Wilson (Wilson 1995), and resource-driven LCSs by Booker (Booker 1999). Each has potential and is worthy of further research; my interest is in what appears to be the most promising of these: Wilson's payoff-accuracy driven XCS model. XCS differs in several ways from more traditional LCSs. The key difference is that in XCS, rule fitness is based on the accuracy of a rule's prediction instead of on the magnitude of the prediction itself. A detailed description can be found in Wilson's original paper on XCS (Wilson 1995); here, I provide an overview of key building blocks in the XCS paradigm.

The Rule Base of an XCS is similar to that of an LCS; however, each classifier has three additional measures: prediction (p), prediction error (ε), and fitness (F). A classifier's prediction, p, is a weighted average of previous rewards; prediction error, ε,

is an estimate of the error in p; and fitness, F, is an inverse function of ε. The GA module uses F for rule generation, while the XCS uses p and ε to control rule firing. This is unlike the LCS model where classifier strength controls all behavior.

Section 4: The Test Bed—The Iterated Prisoner's Dilemma Problem—and a Rationale for its Use

Because the GA paradigm dates back to the 1960s, much progress toward a theory of robust GA design has been made. Although some controversy remains, theoretical work by David Goldberg and his students and co-researchers (Goldberg 2002) has done much to further understanding of GA's internal workings. Because interest in LCS intensified only in the late 1990s, considerable work remains to be done to understand LCS systems and variants like the XCS model. My proposed effort is intended to narrow this gap and to supplement ongoing efforts in this direction by individuals like Riolo, Booker, Wilson, Forrest, and others. With this motivation in mind, the remainder of this section describes the setting for my experimental investigations—the Iterated Prisoner's Dilemma game—and justifies its choice.

In the classic version of the Prisoner's Dilemma (Meng and Pakath 2001), two players compete against each other, and have the choice of either cooperating (C) with, or defecting (D). The payoff obtained by each depends on what action (C or D) each takes. If both players cooperate, each receives a reward of R2. If both players defect, each receives a relatively smaller reward of R3. If one player defects while the other player cooperates, the cooperating player gets a sucker's payoff of R4 while the defector gets the highest possible payoff for the game, R1. The payoffs have the following properties: $R1 > R2 > R3 > R4$ and $(R1 + R4)/2 < R2$.

The first expression states that if both players defect, each does worse than if both cooperate (i.e., $R2 > R3$). Thus, mutual cooperation is preferred to mutual defection. Nonetheless, a player is tempted to defect for two reasons. First, if the opponent were to cooperate, and thus be suckered, the defector's payoff would be greater than with mutual cooperation (i.e., $R1 > R2$). Secondly, even if the opponent defects, the payoff with mutual defection, R3, is better than that if he were to be suckered (i.e. $R3 > R4$). The second expression stipulates that the payoff obtained through unsynchronized alterations of cooperation and defection should not be, on average, better than that obtained through repeated cooperation.

While mutual cooperation could yield a relatively higher payoff of R2, two rational players would both tend to defect (resulting in each earning a smaller payoff of R3). Therein lays the dilemma. The model is traditionally viewed as a useful tool for studying conflicts between self goals and group goals in an organizational or societal setting such as might occur during an "arms race" between nations in a region, or between criminals jointly involved in the commission of a crime. An extension of this scenario is the Iterated Prisoner's Dilemma (IPD) game where a player has multiple "encounters" with another, over time. Each encounter involves a move and a countermove by the two players. Here, a player has the choice of exploiting prior experience in plotting moves in subsequent encounters. An IPD "tournament" consists of a series of IPD games, played with a single or multiple opponents.

The IPD game setting is interesting because it brings the following flexibilities to my setting. First, it allows me to create test situations with various types of characteristics. For instance, one may examine learning and related issues with LCS and XCS (and other types of learning systems) by pitting each against an opponent who uses a deterministic, fixed strategy such as "always defect." Such an opponent (labeled DDD) enables me to study a single-step problem as LCS/XCS cannot initiate behavioral change in DDD. On the other hand, one may pit the system against an opponent who is cooperative as long as its opponent is cooperative, but repays every defection with a defection in the following move (Tit-for-Tat (TFT)). Thus, TFT is a deterministic, reactive player. In this situation, the LCS/XCS must recognize that current action has future (multi-step) ramifications. One may create longer term impacts as with TFTT where the opponent returns two, successive defections in response to one defection by the LCS/XCS. Many such opponents may be easily cast and a few are described in Meng and Pakath (Meng and Pakath 2001).

Second, the IPD setting also allows me to introduce noise into the interactions, an issue that has received little research attention with XCS systems. For example, one may define an opponent called HTFT which is TFT-like but occasionally (i.e., with some predefined probability) turns "hostile" and defects when TFT recommends cooperation (a stochastic, reactive player). The LCS/XCS must learn to cope with such "unexpected" behavior to be successful.

Finally, the IPD setting allows me to determine whether the LCS/XCS can cope with stimuli from multiple opponents. Groups of opponent players may take turns at interacting with the system or may simultaneously interact with it through multiple effectors and detectors. These situations bring up interesting and challenging system architecture-related issues that, to the best of my knowledge, have not been addressed by other researchers.

Section 5: Research Design

Unlike the traditional strength-based LCS model, XCS is accuracy-based. I wish to compare and contrast the two models under different IPD tournament settings to better understand their behaviors. I described the perceived drawbacks of the LCS model in Section 3. Despite such arguments, traditional LCS-based systems have been shown to perform well in some settings, such as evolving novel fighter aircraft maneuvering patterns (Smith, Dike et al. 2000; Smith, Dike et al. 2000) Smith, et al. opine, among other things, that their success is likely due to the emphasis on novelty, rather than control, in their work. Also intriguing is the fact that contrary to conventional wisdom and practice, Smith et al. replace the entire population of classifiers with newly created ones each time the GA module was invoked, without significant performance degradation. Thus, it would appear that the traditional LCS model is not without merit in the appropriate setting. Further, I believe it is also possible that XCS has its own Achilles heel that has not yet been exposed in previous experimentation. Such considerations, coupled with the flexibilities that the IPD game setting offers, motivate me to compare and contrast the LCS and XCS paradigms in that setting. I describe some of the many possibilities below.

5A. Experiment Set A

This experiment set is motivated by the observation (Wilson 1995) that, all else being equal, over-general, low accuracy classifiers tend to proliferate in payoff magnitude-driven (i.e., LCS) systems and to a lesser degree in payoff-accuracy driven (i.e., XCS) systems. This leads me to ask to what extent generalization helps or hinders the former in different IPD contexts? The use of the "#" symbol as part of the condition-side expression of classifiers was first articulated by Holland. Since then, most LCS implementations take its use for granted.

I would like to run two traditional—i.e., payoff-magnitude driven—LCSs on the same task sets (i.e., different IPD tournaments). LCS A-1 will permit the traditional, ternary vocabulary (i.e., 1, 0, and #) in the condition portion of each classifier. LCS A-2 will restrict the conditions to be fully specified (i.e., will only permit 1 and 0). Other than this, the two LCSs will be identical in features, capabilities, starting population sets and fitnesses, opponents encountered, and tournament format (e.g., round-robin, random, at the same time). In particular, (i) both LCSs will use the traditional approach where classifier strength (fitness) is directly tied to payoff magnitude, (ii) both LCSs will use payoff-sharing and (iii) both LCSs will rely on niche GAs for procreation.

Through simulation experiments (the approach traditionally used by the GA and LCS research communities) I intend to contrast the two LCSs in terms of measures such as quality of evolved classifier sets at convergence, number of generations to convergence, evolutionary path traces, relative reward accumulation in the tournaments, etc., and related statistics. These comparisons will be based on appropriate simulation design (e.g., appropriate use of multiple, independent random number streams, adequate number of independent run replications) and sound statistical methodology.

5B. Experiment Set B

The objective of these experiments is the same as that for Set A except that my focus is on contrasting two payoff accuracy-driven LCSs (XCS B-1 and XCS B-2) with each other. Both XCSs will embody the architectures advocated by Wilson for his XCS system, except for the difference in vocabulary used to express classifiers in each. My focus here is only on considerations unique to Set B. The implementation of a payoff accuracy mechanism requires me to define parameters like prediction (p), prediction error (ε), and fitness (F) associated with each classifier. Thus, the goal of the system is to self evolve to a set of classifiers that work well against the IPD Tournament opponents used, without an exhaustive search and without human intervention during the evolutionary process.

5C. Experiment Set C

This set will contrast the two LCS models from Set A with the two XCSs from Set B. How do the four systems stack up against one another? Additional experimentation will not be necessary but comparisons and related statistical procedures, plots, and traces will have to be executed.

5D. Experiment Set D

This set is motivated by the following consideration: The systems in Experiment Sets A and B react, or self-adjust, based on feedback for each individual action that each performs. What would be the impact of allowing these systems to self-adjust on collective feedback for a series of actions instead? The collective feedback could be in the form of an aggregate reward measure such as total reward or average reward for all actions performed collectively.

The implications of this shift in focus when playing against a group of opponents instead of a single opponent are interesting. The system would be judged not based on its performance against each individual but against the entire group. Also interesting are the implications in tournaments against multi-step opponents where the payback for a current move could come many moves later or be spread over multiple subsequent moves. I believe that this change in feedback poses interesting, as-yet unexplored, challenges to both types of systems.

To enable this set, I will have to institute one change to the classifiers used. The right hand sides of each will no longer advocate a single action but a series of actions. Also, the GA module will be able to apply crossover and mutation to the action sequences apart from the memory sequence (the left hand sides of the classifiers). Consequently, the initial population generation mechanism will also need re-thinking.

Note that the resultant LCS and XCS systems would be "planning systems." From this perspective, the question I am posing is, "to what extent can each type of system plan an optimal sequence of actions based on prior experience with an opponent or opponent group?"

Conclusion

The key motivation for this effort is to further the understanding of LCS- and XCS-based learning mechanisms. Previous studies of GA, LCS, and XCS systems have involved significant amounts of trial-and-error and intuition. Therefore, one of the aims of the proposed research is to draw theoretical insights which might lead to a better understanding of how such systems work and the ability to design more robust systems which require less trial-and-error.

References

Booker, L. B., Ed. (1999). *Do We Really Need to Estimate Rule Utilities in Classifier Systems? Learning Classifier Systems: From Foundations to Applications*, Berlin: Springer-Verlag.

Booker, L. B., D. E. Goldberg, et al. (1989). "Classifier Systems and Genetic Algorithms." *Artificial Intelligence* **40**: 235-282.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons From and For Competent Genetic Algorithms,* Norwell, MA: Kluwer Academic Publishers.

Jantke, K. P. and S. Lange (1995). *Algorithmic learning for knowledge-based systems : GOSLER final report*. Berlin: Springer-Verlag.

Meng, C.-L. and R. Pakath (2001). "The Iterated Prisoner's Dilemma: Early Experiences with Learning Classifier System-based Simple Agents." *Decision Support Systems* **31**(4): 379-403.

Riolo, R. L. (1991). *Lookahead Planning and Latent Learning in a Classifier System. Simulation of Adaptive Behavior*, Cambridge, MA: MIT Press.

Smith, R. E., B. A. Dike, et al. (2000). "Classifier Systems in Combat: Two-sided Learning of Maneuvers for Advanced Fighter Aircraft." *Computer Methods in Applied Mechanics and Engineering* (186).

Smith, R. E., B. A. Dike, et al. (2000). "The Fighter Aircraft LCS: A Case of Different LCS Goals and Techniques." *Learning Classifier Systems: From Foundations to Applications*. P. L. Lanzi, Berlin: Springer-Verlag.

Wilson, S. W. (1995). "Classifier Fitness Based on Accuracy." *Evolutionary Computation* **3**(2): 149-175.