

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems
(AMCIS)

December 2004

Alternate Adaptive Agent Architectures and Behavioral Consequences

Dave Gaines
University of Kentucky

Ramakrishnan Pakath
University of Kentucky

Follow this and additional works at: <http://aisel.aisnet.org/amcis2004>

Recommended Citation

Gaines, Dave and Pakath, Ramakrishnan, "Alternate Adaptive Agent Architectures and Behavioral Consequences" (2004). *AMCIS 2004 Proceedings*. 5.
<http://aisel.aisnet.org/amcis2004/5>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Alternate Adaptive Agent Architectures and Behavioral Consequences

David Gaines
University of Kentucky
dgaines@uky.edu

Ramakrishnan Pakath
University of Kentucky
pakath@uky.edu

ABSTRACT

The Learning Classifier System (LCS) and its descendant, XCS, are promising paradigms for adaptive agent construction. Whereas LCS allows classifier payoff predictions to guide system performance, XCS focuses on payoff-prediction accuracy instead, allowing it to evolve “optimal” classifier sets in particular applications requiring rational thought. We examine LCS/XCS performance in artificial situations with broad social/commercial parallels, created using the non-Markov Iterated Prisoner’s Dilemma (IPD) game-playing scenario, where the setting is sometimes asymmetric and where irrationality sometimes pays. We systematically perturb a “conventional” IPD-playing LCS-based agent until it results in a full-fledged XCS-based agent, contrasting the simulated behavior of each LCS variant with the XCS agent in terms of a number of performance measures. Our intent is to examine the XCS paradigm to understand how it better copes with a given situation (if it does) than the LCS perturbations studied.

Keywords

Adaptive Agents, Genetic Algorithms, Learning Classifier Systems, XCS, Iterated Prisoner’s Dilemma

INTRODUCTION

The Learning Classifier System (LCS) is an early paradigm for evolutionary computation (Booker, Goldberg and Holland, 1989). An LCS uses limited-vocabulary, fixed-length rules (classifiers), does not require associated utility/fitness values to be pre-specified or remain stationary, accommodates parallel rule firing, and can autonomously spawn syntactically-correct potentially-more-useful rules from existing ones using an embedded genetic algorithm (GA). The model has seen wide application including data monitoring, mining, and aircraft maneuvering (Bonelli, Parodi, Sen and Wilson, 1990, Browne, Holford, Moore and Bullock, 2000, Flockhart and Radcliffe, 1996, Holmes, Durbin and Winston, 2000, Saxon and Barry, 1999, Smith, Dike, Mehra, Ravichandran and El-Fallah, 2000) with variable success. Further scrutiny identified several drawbacks arising from using a classifier’s fitness to measure both its genetic utility and its payoff prediction/firing likelihood (Wilson, 1995):

- Classifiers operating in higher-payoff problem-space niches could dominate by growing disproportionately fitter over time.
- To mitigate the above, some implementations share reward with all fireable classifiers that advocate the same action as the firing classifier. Each classifier’s (relatively lower) fitness becomes a poorer predictor of its payoff.
- Early rules in a chain appear less fit over time, even with reward back-propagation. A solution is to use “niche GAs” (that operate in population niches) as opposed to panmictic GAs (that operate on the entire population). Even so, two problems remain ...
- The LCS is unable to distinguish between more specific and more general (having more # symbols) classifiers having a certain payoff expectation. The general versions match more environmental states and tend to proliferate.
- LCS’s have no mechanism to ensure accurate (low payoff-prediction-error rate) generalizations.

The most promising refinement of LCS, called XCS (Wilson, 1995), uses fitnesses only for rule generation, while payoff prediction and prediction error are used to control firing. In principle, XCS’s strengths include its ability to generate “maximally general” classifiers (further generalizations without accuracy losses are impossible) and state-action-payoff mappings that are “complete” (each state-action pair is represented), “accurate” (each classifier maps to a single payoff), “non-overlapping” (no classifier is duplicated), and “minimal” (only classifiers meeting the preceding conditions are present). Consequently, XCS’s knowledge base is more comprehensible, more compact, and is potentially more useful in solving problem perturbations.

LITERATURE REVIEW

XCS-based systems that are more effective than LCS-based ones have been created for the Boolean multiplexer, maze navigation, gaming, and data mining problems (Bernadó-Mansilla and Garrell-Guiu, 2003, Butz, Sastry and Goldberg, 2002, Falke and Ross, 2003, Kovacs, 1997, Kovacs and Kerber, 2001, Lanzi, 1997, Wilson, 1995). However, particular LCS-based systems perform surprisingly well (Smith et al., 2000). Thus, the LCS is not entirely without merit; nor can one regard the XCS as a panacea. Further, such prior LCS/XCS investigations involve systems capable of rational thought.

Such considerations prompt us to contrast the two paradigms using a common test-bed – the Iterated Prisoner’s Dilemma (IPD) game-playing scenario. This setting is inherently non-Markov, sometimes asymmetric, and one where irrationality sometimes outperforms rationality -- characteristics that make it particularly challenging to an artificial player. The IPD game is also one with broader commercial/social parallels than prior settings explored. Although it has received sustained research scrutiny since the fifties, research momentum exploded after Axelrod’s (1984) pioneering efforts in applying evolutionary systems to outwit humans. The impetus continues; the Engineering and Physical Sciences Research Council, U.K., is currently co-sponsoring a follow-up to Axelrod’s IPD competitions. To us, the IPD game is a very useful test-bed for studying adaptive agent behavior. By “adaptive agent” we mean a special purpose, intelligent software entity that seeks to autonomously evolve its knowledge state over time to better cope with a situation. We believe that the LCS/XCS models are deserving of continued research scrutiny as feasible means of constructing adaptive agents. Prior research (e.g., ref. 5 and 6) documents the design and functioning of agents for a variety of tasks. Four features distinguish our work: we consider agents with possible interest conflicts; we assume agent strategies are not common knowledge; agent behavior is learned; learning is autonomous. Consider a home buyer and broker (Meng and Pakath, 2001). The buyer may not be up-front with the broker in regards to his/her expectations and intent/ability to purchase. The broker may not favor the buyer’s best interests as the commission is paid entirely by a seller. Both may harbor other such biases. A buyer-broker pair learns more about one another’s operational strategies through mutual interaction. This scenario extends to web-based transactions where buyers, brokers, and sellers deploy adaptive agents that must discern one another’s strategies through interaction. Because any "opponent" is an "unknown," canning an effective buyer/broker/seller agent beforehand becomes difficult/impractical. The situation becomes more complex when an agent must deal with several others as when a buyer agent must deal with several brokers, each representing several sellers and multiple commodities. Where canning effective strategies is impractical for any reason, one may attempt creating agents that autonomously adapt strategies.

In the Prisoner's Dilemma (PD), two players can mutually cooperate (C) or defect (D). If both cooperate or defect, each receives a reward of R_2 or R_3 , respectively. If one defects while the other cooperates, the latter gets a sucker’s payoff of R_4 while the former gets R_1 . Here, $R_1 > R_2 > R_3 > R_4$ and $(R_1 + R_4)/2 < R_2$. Thus, while mutual cooperation is preferred to mutual defection ($R_2 > R_3$), individual defection is tempting ($R_1 > R_2$; $R_3 > R_4$), and, repeated cooperation is more lucrative than each alternately playing sucker. The dilemma: Must a player cooperate or defect? In an Iterated PD (IPD), players repeatedly play one another and could exploit prior experience. An IPD "tournament" consists of a series of IPD games against a single or multiple opponents. The IPD scenario is an excellent LCS/XCS-based adaptive agent test bed because:

--A variety of “opponent” strategies ranging from consistent cooperation/defection (CCC/DDD) to completely random/chaotic behavior (RAND) can be defined; the LCS/XCS must discern such strategies.

--In some instances, irrational behavior nets greater rewards, a characteristic shared by many business/social contexts. E.g., with TFT (Tit-for-Tat) who mirrors an agent’s current play in the next encounter, repeated cooperation is best.

--IPD is inherently non-Markov; more-informed action choices call for agent memory enabling us to test various memory architecture (size and content) designs.

--Some situations result in asymmetric knowledge base updates because of unequal coverage of the input domain by detected stimuli, potentially negatively impacting agent learning/performance. E.g., a DDD opponent (consistent defector) never exposes the agent to cooperation.

--We can create both immediate reward and various deferred reward situations. E.g., a TFT opponent results in a single one step-delayed feedback while TTFT (Two-Tit-for-Tat) results in two consecutive feedbacks after a one-step delay.

--We could introduce noise. E.g., HTFT is TFT-like but occasionally turns “hostile” and defects when TFT recommends cooperation.

--We can simulate multiple, confounding stimuli; diverse opponents may simultaneously interact with an agent.

Thus, a single game-playing setting allows us to assess our adaptive agent paradigms in a variety of real world-relevant ways.

METHODOLOGY

We compare and contrast LCS and XCS under specific IPD tournament settings to: (a) better understand their adaptive behavior and (b) determine to what extents the virtues of XCS hold in more complex settings as exemplified by the IPD.

Using simulation experiments, we repeatedly compare learning and steady-state behavior characteristics of a modern IPD-playing XCS-based adaptive agent (XCS) with that of a series of LCS-based agents beginning with a “traditional” model (T-LCS) and systematically perturbing T-LCS, bringing it closer in features and functions to XCS, until it coincides with XCS. In each comparison, both agents play against the same IPD opponent(s). Our approach draws on key architectural differences between XCS and T-LCS.

Characteristic	T-LCS	XCS
Classifier Fitness Determinant	Prediction Magnitude	Prediction Accuracy
Initial Population Generation	Random	Through Covering
Population Size	Constant, N	$\leq N$
Genetic Algorithm	Panmictic	Niche
Parent Selection	Fitness Proportional	Tournament
Classifier Deletion Criteria	Classifier Fitness	Classifier Fitness and Resource Balancing
Action Selection	Fitness Proportional	Biased Exploration
Classifier Fitness Updates	Firing Classifier	All Matching Classifiers advocating the same Action

Table 1. Key Architectural Differences

The impacts of individual perturbations to T-LCS on performance may be sensitive to the ordering of perturbations. We introduce perturbations one at a time initially, observe the impacts of each, and then create appropriate perturbation combinations and sequences from the results.

Performance Measures

Each agent goes through a “learning phase” and a “steady-state/performance phase.” We use the following specific measures:

Performance: The proportion of correct agent responses given the last x inputs (Wilson, 1995).

Learning Rate: Number of iterations to achieve $y\%$ of steady-state performance normalized by the latter to yield a “buck/bang” learning ratio (Holmes, 1998, Wilson, 1995).

Problem Difficulty: The proportion of the “optimal” (i.e., accurate, minimal, non-overlapping, and complete) population for a problem, in the knowledge base (Kovacs and Kerber, 2001).

Evolutionary Path Traces: Trace a classifier’s history since inception (Wilson, 1995).

Relative Earnings: The relative total reward accumulated by an agent vis-à-vis its opponent(s) (Meng and Pakath, 2001).

Statistical Measures: Given multiple IID simulation run replications, we use parametric and/or non-parametric multivariate statistical techniques to compare a vector of multiple performance measures at a given epoch for XCS with that /those for an/several LCS variant(s). (The prevalent approach is to graph relative performances for individual measures.)

Hypotheses

Test hypotheses are based on our expectations regarding an LCS variant’s relative performance vis-à-vis XCS against a particular opponent. The extents to which XCS and LCS could accurately generalize depend on the opponents involved; we expect XCS to perform significantly better in situations that do permit accurate generalizations. For a given opponent, we expect the performance differential between XCS and LCS to narrow as their architectural differences narrow. We explore

relative performance against RAND, CCC, DDD, TFT, TTFT, and HTFT. With RAND, prior knowledge should offer no future insights. Both CCC and DDD should cause an agent to consistently defect but for different reasons. TFT should propel an agent toward mutual cooperation. TTFT should do the same but at a faster rate. HTFT's behavior is a function of its defection probability, p (e.g., HTFT=TFT if $p=0$; HTFT=DDD if $p=100$). We hope to extend such explorations in other meaningful ways in the longer term.

RESULTS AND CONCLUSION

We provided a brief overview of a simulation study seeking to contrast two adaptive agent paradigms – the LCS and the XCS. We draw upon extant research to make a case for more exhaustive testing of both models and argue why the IPD is an ideal test bed with high business/social relevance. We then described our experimental purpose and methodology. Our Research-in-Progress presentation will encompass the above issues, provide further experiment operationalization details, and present and discuss simulation results and hypotheses test findings. We hope the finished work will help us better understand specific features of LCS and XCS that are particularly valuable in the generic IPD setting and that our efforts will add to the growing body of knowledge on adaptive systems in general and the LCS and XCS paradigms in particular.

ACKNOWLEDGEMENTS

This research work was supported in part by a grant from the Kentucky Science and Engineering Foundation as per Grant Agreement #KSEF -148-502-04-110 with the Kentucky Science and Technology Corporation.

REFERENCES

1. Axelrod, R. (1984) *The evolution of cooperation*, Basic Books, New York.
2. Bernadó-Mansilla, E. and Garrell-Guiu, J. M. (2003) Accuracy-based learning classifier systems: models, analysis and applications to classification tasks, *Evolutionary Computation*, 11, 3, 209-238.
3. Bonelli, P., Parodi, A., Sen, S. and Wilson, S. W. (1990) NEWBOOLE: a Fast GBML System, *Proceedings of the Seventh International Conference on Machine Learning*, Austin, TX, 153-159.
4. Booker, L. B., Goldberg, D. E. and Holland, J. H. (1989) Classifier Systems and Genetic Algorithms, *Artificial Intelligence*, 40, 235-282.
5. Browne, W., Holford, K., Moore, C. and Bullock, J. (2000) An Industrial Learning Classifier System: The Importance of Pre-Processing Real Data and Choice of Alphabet, *Engineering Applications of Artificial Intelligence*, 13, 1, 25-36.
6. Butz, M. V., Sastry, K. and Goldberg, D. E. (2002) Tournament Selection in XCS, Illinois Genetic Algorithms Laboratory Report No, 2002020, Urbana, IL, 1-17.
7. Falke, W. J. I. and Ross, P. (2003) Dynamic Strategies in a Real-Time Strategy Game, *Proceedings of the Genetic and Evolutionary Computation Conference 2003*, Chicago, IL, 1920-1921.
8. Flockhart, I. W. and Radcliffe, N. J. (1996) A Genetic Algorithm-Based Approach to Data Mining, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, OR, 299-302.
9. Holmes, J. H. (1998) Differential Negative Reinforcement Improves Classifier System Learning Rate in Two-Class Problems with Unequal Base Rates, *Genetic Programming 1998*, 635-642.
10. Holmes, J. H., Durbin, D. R. and Winston, F. K. (2000) A New Bootstrapping Method to Improve Classification Performance in Learning Classifier Systems, *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature*, Paris, France, 745-755.
11. Kovacs, T. (1997) XCS Classifier System Reliably Evolves Accurate, Complete, and Minimal Representations for Boolean Functions, *Soft computing in engineering design and manufacturing*, Springer Verlag, Berlin, 59-68.
12. Kovacs, T. and Kerber, M. (2001) What Makes a Problem Hard for XCS?, *Advances in learning classifier systems 1996*, Springer-Verlag, Berlin, 80-99.
13. Lanzi, P. L. (1997) A Study of the Generalization Capabilities of XCS, *Proceedings of the Seventh International Conference on Genetic Algorithms*, San Francisco, CA, 418-425.
14. Meng, C.-L. and Pakath, R. (2001) The Iterated Prisoner's Dilemma: Early Experiences with Learning Classifier System-based Simple Agents, *Decision Support Systems*, 31, 4, 379-403.
15. Saxon, S. and Barry, A. (1999) XCS and the Monk's Problems, *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, FL, 13-17.
16. Smith, R. E., Dike, B. A., Mehra, R. K., Ravichandran, B. and El-Fallah, A. (2000) Classifier Systems in Combat: Two-sided Learning of Maneuvers for Advanced Fighter Aircraft, *Computer Methods in Applied Mechanics and Engineering*, 421-437.
17. Wilson, S. W. (1995) Classifier Fitness Based on Accuracy, *Evolutionary Computation*, 3, 2, 149-175.