LEARNING CLASSIFIER SYSTEMS
Poster Papers

Larry Bull, chair

Towards the Use of XCS in Interactive Evolutionary Design

Larry Bull
Faculty of Computing Engineering & Mathematical Sciences
University of the West of England
Bristol BS16 1QY

David Wyatt
Faculty of Computing Engineering & Mathematical Sciences
University of the West of England
Bristol BS16 1QY

Ian Parmee
Faculty of Computing Engineering & Mathematical Sciences
University of the West of England
Bristol BS16 1QY

Abstract

Learning classifier systems represent a technique by which various characteristics of a given problem space may be deduced and presented to the user in a readable format. W present results from the use of XCS on simple tasks with the general multi-variable features typically found in problems addressed by an Interactive Evolutionary Design process. That is, we examine the behaviour of XCS with versions of a well-known single-step task and consider the speed of learning, noise, and the ability to respond to changes. We introduce a simple form of supervised learning for XCS with the aim of improving its performance with respect to these two measures. Results show that improvements can be made under the new learning scheme.

1 sXCSR

Interactive Evolutionary Design (IED) (Parmee and Bonham, 1999) moves away from the use of evolutionary computing techniques within a rigid optimization environment and instead utilizes them as generators and gatherers of optimal design information. The approach involves the capture of designer experiential knowledge and intuition within adaptive search processes through an iterative designer/machine-based refinement of the design space. This last aspect of the process is of interest to us here: we consider a way in which to enhance the presentation of results from a given iteration of the search process through the use of learning classifier systems.

XCS (Wilson, 1995) has been shown to perform well on a number of benchmark data mining tasks with the added benefit of producing readable production system rules. XCS uses the incremental Widrow-Hoff procedure to update expected payoff values. Here, we introduce a simplified update procedure whereby newly created rules, i.e. those which have never participated in an action set since their creation (via cover or the GA), have their expected payoff value set to that of the first training instance they experience. This value remains constant. All other parameters are initialized and updated as in traditional XCS. With a real-numbered representation scheme this system is here termed sXCSR. The system was applied to a single-step function defined for binary strings of length $l = k + 2^k$, that is, a real-numbered multiplexor problem. Figure 1 shows that improvements can be made under the new learning scheme.

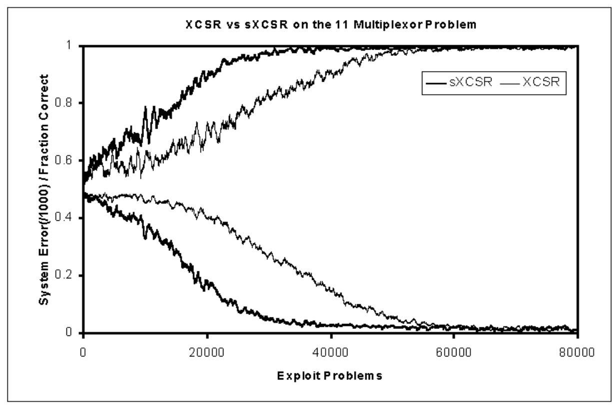

Figure 1: XCSR and sXCSR on the 11-variable multiplexer task.

In terms of the introduction of a learning technique to the Interactive Evolutionary Design concept, this work can only be considered a preliminary investigation. However, the results strongly indicate a significant potential in the utilisation of learning classifier systems to support designers as part of the IED process. Utility could extend beyond this initial task to generating rules relating to a wide spectrum of relevant design information.

References

Wilson, S. (1995) Classifier fitness based on accuracy,. *Evolutionary Computation*, 3(2), 149-175.

Parmee I. & Bonham C. (1999) Towards the Support of Innovative Conceptual Design Through Interactive Designer / Evolutionary Computing Strategies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing Journal*; 14, pp. 3–16.

Bull, L., Wyatt, D. & Parmee, I. (2002) Towards the Use of XCS in Interactive Evolutionary Design. *UWE Learning Classifier Systems Technical Report UWELCSG02-003* – http://www.cems.uwe.ac.uk/lcsg.

An Experimental Comparison of Genetic and Classical Concept Learning Methods

Gabriella Kókai[1]
[1]Department of Programming Languages
Friedrich-Alexander University
Martensstr. 3. D-91058 Erlangen, Germany
e-mail: kokai@informatik.uni-erlangen.de

Zoltán Tóth[1,2], **Szilvia Zvada**[1]
[2]Department of Informatics
University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
e-mail: {zntoth, zvada}@inf.u-szeged.hu

1 INTRODUCTION

In this work the classical learning methods *C4.5* [Qui93] and *FOIL* [Qui90] are compared with the genetic learning systems *GEA* (Generic Evolutionary Programming Library, [Tot01]) and *GeLog* (Genetic Logic Programming, [Kok01]). Two problems were involved in the comparison: A mushroom classification and a chess endgame problem. The experiences show that the evolutionary methods not only reach the performance of the traditional learning systems but in complex tasks even outperform them.

2 THE TESTED SYSTEMS

GEA is an evolutionary optimizer tool, with implementation of *EAs* and *ESs*. Due to the applied plug-in technology, it is easily extendible with new individual representation forms and evolutionary algorithms.

In contrast to *GEA*, the *GeLog* system alloys the robust searching technique of genetic algorithms with the learning approach of *ILP* (inductive logic programming). It evolves Prolog programs by means of background knowledge and positive and negative examples.

FOIL is a relational learning system developed in the framework of ILP. It learns Horn clauses by considering their coverage on the training data which is expressed as relations.

The last system partaken in the comparison is the decision tree learner *C4.5*. It builds decision trees by the guidance of information-based heuristics from the attribute-value representation of the training examples.

3 THE LEARNING TASKS

The first learning task in the comparison is the recognition of poisonous mushrooms from 22 attributes, sometimes with missing values. The database contains the descriptions of 8124 mushrooms, 48.2% of which are poisonous. In the second task, the goal is to classify board configurations for the chess endgame situation *white king and rook against black king* (KRK). A configuration is considered positive if white wins immediately or the result of the game is draw. 10.1% of the 28056 examples in the database are positive.

4 RESULTS

All of the tested systems achieved basically the same classification accuracy on the simpler mushroom classification problem. On the more complicated KRK problem, the genetic learners overperformed *C4.5* and produced rule sets that contain significantly less rules than the results of both classical greedy learning systems. The price of the simpler and more general hypotheses is the longer execution time of the learning processes.

References

[BM98] Blake, C. L., Merz, C. J.: *UCI Repository of Machine Learning Databases.* http://www.ics.uci.edu/~mlearn/MLRepository.htm

[Kok01] Kókai, G.: *GeLog - A System Combining Genetic Algorithm with Inductive Logic Programming.* In Proceedings of the 7th Fuzzy Days, Springer-Verlag pp. 326–344 LNCS 2206, 2001

[Qui90] Quinlan, J. R.: *Learning Logical Definitions from Relations.* Machine Learning, 5 pp. 239–266, 1993

[Qui93] Quinlan, J. R.: *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993

[Tot01] Tóth, Z., Kókai, G.: *An Evolutionary Optimum Searching Tool.* in Proceedings of IEA/AIE-2001, Springer-Verlag LNAI 2070 pp. 19–24, 2001

Cooperative concept learning by means of a distributed GA

Filippo Neri
Marie Curie Fellow at Unilever Research Port Sunlight
Quarry Road East, Bebington, Wirral, CH363JW, UK, and
Researcher at University of Piemonte Orientale, Italy
email: neri@di.unito.it

1 Extended abstract

Two cooperative concept learning strategies are investigated with respect to the features of the found concept descriptions. The system REGAL is used as a experimental framework. The objective is to produce a more efficient learning system. A description about how to setup a suitable experimental setup is reported.

It is worthwhile to note that, in principle, these cooperative learning strategies could be applied to a pool of different learning systems.

REGAL [Neri and Saitta, 1996, Neri, 1997] learns relational disjunctive concept descriptions in a restricted form of First Order Logic by using cooperative evolution. REGAL's architecture is a network of N processes $GALearners$, coordinated by a $Supervisor$ that imposes cooperation among the evolving populations. Each $GALearner_n$ tries to find a description for a subset of the learning instances LS_n by evolving its population. In addition, the $GALearners$ may perform migration (exchange) of individuals. The $Supervisor$ coordinates the distributed learning activity by periodically assigning different subsets of the learning instances to the $GALearners$. The composition of these subsets depends on the specific cooperative policy used. Two policies of cooperation have been investigated.

As no a priori information is available on what is a successful assignment of learning instances, we decided to develop two cooperative learning strategies based on different assumptions.

The first cooperative learning strategy, named Let Seed Expand, works as follows: when a learner find a description ψ, remove from its learning set all the instances covered by other already found descriptions and not covered by ψ, and let ψ improve. In some sense, this policy realizes a pool of "divide et impera" learners evolving in parallel.

The second form of cooperation, named Describe Those Still Uncovered, forces the learners in dealing as soon as possible with the instances difficult to cover. Essentially, as soon as a promising concept description emerges, the instances not covered by it are included into all the learning sets, whereas each covered instance is inserted into only one learning set.

The two cooperative strategies show different behaviors with respect to the features of the found concept descriptions. We believe that a (distributed genetic base) learner able to exploit both cooperative strategies may acquire satisfactory concept descriptions across a wide range of applications. Further research to investigate this claim is in progress.

References

[Neri, 1997] Neri, F. (1997). *First Order Logic Concept Learning by means of a Distributed Genetic Algorithm.* PhD thesis, Department of Computer Science. University of Torino, Italy.

[Neri and Saitta, 1996] Neri, F. and Saitta, L. (1996). Exploring the power of genetic search in learning symbolic classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-18:1135–1142.