COGIN: Symbolic Induction with Genetic Algorithms

David Perry Greene and **Stephen F. Smith**
Graduate School of Industrial Administration / School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
dg1v@andrew.cmu.edu and sfs@isl1.ri.cmu.edu

Abstract

COGIN is a system designed for induction of symbolic decision models from pre-classed examples based on the use of genetic algorithms (GAs). Much research in symbolic induction has focused on techniques for reducing classification inaccuracies that arise from inherent limits of underlying incremental search techniques. Genetic Algorithms offer an intriguing alternative to stepwise model construction, relying instead on model evolution through global competition. The difficulty is in providing an effective framework for the GA to be practically applied to complex induction problems. COGIN merges traditional induction concepts with genetic search to provide such a framework, and recent experimental results have demonstrated its advantage relative to basic stepwise inductive approaches. In this paper, we describe the essential elements of the COGIN approach and present a favorable comparison of COGIN results with those produced by a more sophisticated stepwise approach (with support post processing) on standardized multiplexor problems.

Introduction

The conventional symbolic induction problem is one of formulating a decision model which maps combinations of attribute-values into a set of classes. Given a set of pre-classed examples, the goal is to find the "simplest" model which most accurately classifies the data. To deal with the combinatorics of constructing symbolic models, traditional techniques rely on a deterministic stepwise bias. While this bias makes search tractable, it is acknowledged that in noisy or complex spaces, such an approach can result in misformed models which predict poorly [Breiman et.al. 1984]. Recognizing this, a large body of research has focused on rectifying this problem through the addition of auxiliary techniques (e.g., pruning[Quinlan 1986], model transformation[Quinlan 1987], alternative fitness measures[Holte, Acker & Porter 1989], or representation change[Matheus & Rendell 1989]). In many cases these corrective techniques have shown very good results.

An alternative approach to dealing with the combinatorics of constructing symbolic decision models is to move outside of the confines of stepwise search paradigms. Genetic Algorithms (GAs) offer one such possibility, which has motivated the development of COGIN, a GA-based system designed specifically for symbolic induction from examples. In a recent study, initial experiments with COGIN demonstrated the clear superiority of it's genetic based search over basic stepwise approaches on problems of increasing complexity.

A recent study by Quinlan [Quinlan 1988] presents an opportunity to compare the performance of the COGIN approach versus a post-processing alternative. In his paper, Quinlan provided comparisons with an earlier genetic classifier on the statistical multiplexor problem. The multiplexor is a particularly interesting function because it provides the type of non-linear complexity which is problematic for decision tree building. In this paper we describe the conceptual framework underlying the COGIN approach and examine three multiplexor functions of increasing scale complexity under conditions of varying training samples sizes (as described in Quinlan's original paper).

GAs and Symbolic Induction

In contrast to deterministic, step-wise search mechanisms, the GA is a probabilistic search technique, based on the concept of adaptive efficiency in natural organisms, where solutions are iteratively developed and improved through competition among alternatives [Holland 1975]. A GA maintains a pool of candidate solution structures (the population). A search cycle is iterated where structures in the current population are selected and recombined to produce new structures

(offspring) for evaluation, and a new population is configured. Selection of structures is biased by their evaluated utility (fitness), and new structure creation via recombination acts to exploit the constituent "building blocks" of high performance structures. From a sampling perspective, this leads to an increasing concentration of trials to those regions of the search space observed to be most profitable. By their probabilistic nature, GAs are relatively insensitive to both nonlinearities and noise. The reader is referred to [Goldberg 1989] for an introduction to the theory and practice of GAs.

Previous research in GA-based learning provides two basic paradigms for mapping this basic search model to the symbolic induction problem. The first, designated the Pitt Approach (e.g., [Smith 1983, Grefenstette, Ramsey & Schultz 1990, DeJong & Spears 1991]), assumes a population of alternative decision models. In this case, the fitness measures driving the search relate to the overall worth of candidate models (e.g., classification accuracy). The second paradigm, referred to as the Classifier System (or Michigan) approach (e.g., [Holland 1986, Wilson 1987]), alternatively assumes that the population collectively constitutes the current decision model, with individuals representing specific model components. In this case, fitness measures reflect the utility of individual components within the model (e.g., discriminability), and the overall performance of the model is evaluated externally. In either paradigm, a model representation is required that allows generation of meaningful new structures (models or model components) through simple recombination of existing structures. In the context of symbolic induction this is most simply accomplished by representing a decision model as a disjunctive set of rules, where each rule (i.e. model component) contains a single conjunctive condition over an ordered list of attribute values and a classification.

COGIN

Effective use of genetic search in the context of conventional symbolic induction requires a framework that promotes the fundamental model building objectives of predictive accuracy and model simplicity. Such a framework is provided by COGIN (COverage-based Genetic INduction), a system for genetic induction of symbolic decision models from pre-classed examples[Greene & Smith 1992]. COGIN manipulates a population of single condition rules which are collectively interpreted as a candidate decision model, analogous to the classifier system paradigm mentioned above. However, in direct contrast to the incremental learning assumptions of classifier system research, COGIN is designed to directly exploit the static structure of the conventional symbolic induction problem. In particular, coverage of the examples in the training set is used throughout the search as both an explicit constraint on model complexity (size) and a basis for creating a pressure toward appropriate classification diversity within the model. On any given cycle of COGIN's search, newly created rules compete with rules in the current decision model to cover subsets of the training set, and survival of a given rule is a direct function of its ability to better fill a particular classification niche in the evolving model than any competitors.

In more detail, the operation of COGIN proceeds as follows. At the start of each cycle, a percentage of the rules constituting the current model are randomly paired and recombined to produce a set of new candidate rules. Each of these rules in then assigned a fitness measure and pooled with the original rules of the current model for competitive configuration of a new model. The fitness of a given rule is defined as lexicographically-screened entropy (using the formulation given in [Mingers 1989]). That is, maximum entropy within a given interval of classification error.

The heart of the cycle is the competition step, which is carried out through application of a *coverage-based filter*. The filter operates by "allocating" training examples to rules that match them in fitness order. The output of the filter, which constitutes the new decision model, is the set of rules receiving at least one example, and all other candidate rules are discarded. After the new model has been determined, its predictive accuracy relative to the training set is evaluated. During this process, fitness is used as a basis for conflict resolution, and a default rule (the majority class of the training set) is invoked in the event that no rules match a given training set example. If the new model yields a higher predictive accuracy than the best model evaluated thus far in the search or performs at the same level and contains fewer rules than the previous best model, then it is retained as the new best for future comparison and the cycle repeats. The overall search process terminates after a pre-specified number of iterations, and the best model produced during the search is taken as the final model.

Figure 1 provides a representative example the dynamics of this search process, assuming a training set of 7 examples, with examples 1, 2 and 5 of class 0 and examples 3, 4, 5, and 7 of class 1. Notationally, each rule in the figure is depicted by number, followed by the set of matching examples, its action and its fitness. For example, [r1:1 3 4 6→1(.089)] indicates that rule r1 matches examples 1, 3, 4 and 6, assigns class 1 and has a fitness of .089. At generation N the current model

	Training set size	COGIN		Decision Tree		Derived Rules	
		size	accur.	size	accur.	size	accur.
F6	50	14.7	87.2%	10.2	76.9%	3.0	85.1%
	100	17.6	94.5%	21.8	91.8%	4.0	100%
	150	15.1	97.4%	22.2	98.0%	4.0	100%
	200	14.8	98.1%	24.6	100%	4.0	100%
F11	100	33.1	70.1%	23.0	63.2%	5.8	71.8%
	200	33.4	93.3%	41.8	78.1%	8.2	98.3%
	300	30.1	97.2%	63.4	86.6%	8.0	100%
	400	26.4	99.4%	68.6	92.5%	8.0	100%
	500	23.6	99.9%		***		***
F20	200	73.0	59.7%	49.0	68.8%	8.6	69.2%
	400	122.2	69.0%	95.8	82.1%	14.4	88.0%
	600	137.8	82.4%	121.0	87.4%	16.2	97.4%
	800	113.0	93.1%	171.4	92.4%	18.4	98.3%
	1000	84.1	97.5%		***		***

Table 1: Comparative Performance of COGIN and C4

with C4.5 (a descendant of C4) on the 6 multiplexor problem was reported. Running both systems in so called "batch-incremental" mode, GABIL was found to outperform C4.5 with respect to online predictive performance over time as new examples were sequentially accumulated.

COGIN, on the other hand, is directly based on the same "batch learning" assumptions as C4, and designed specifically to solve the same class of symbolic induction problems. As such, Quinlan's study provides benchmark results against which the performance of COGIN can be directly evaluated, and such an evaluation would appear to provide an isolatable assessment of GA-based and stepwise search procedures in a batch learning context. Moreover, since [Quinlan 1988] reports both the performance results obtained by C4's decision tree building approach (with pruning) and the improved results obtained by C4's additional "decision tree to production rule" post-processing option, this study provides the opportunity to assess the performance of a COGIN's GA-based inductive framework relative to a step-wise inductive framework with various levels of auxiliary support techniques. In the following section, the results of this experimental comparison are reported.

Experimental Results

Table 1 contains performance results obtained with COGIN on the 6, 11, and 20 bit multiplexor problems (denoted F6, F11, and F20 respectively) along with the results previously reported for C4 in [Quinlan 1988]. For each problem instance (F6, F11, or F20), runs were made with five randomly generated training sets of each designated size and performance of the generated decision model was evaluated with respect to a holdout set of 1000 previously unseen cases. Given the stochastic nature of COGIN's search, five runs were made with each generated training set. In each run, COGIN's search was terminated after a pre-specified number of iterations (generations). In the case of the F6 experiments, the search was run for 300 generations; in the F11 and F20 experiments the search was run for 600 generations. The performance results reported in Table 1 for each training set size reflect averages across these runs (25 in the case of COGIN, 5 in the case of the deterministic C4). The average model sizes shown reflect number of rules or number of decision tree nodes as appropriate.

From the results, we see that COGIN performed comparably to C4 across all instances of the multiplexor problem considered. In both the 6 and 11 bit multiplexor problems, COGIN achieves accuracy levels above 90% at the same training set size as is achieved by C4's post generated rules (although at lower absolute levels), and in the largest 20 bit multiplexor problem an accuracy level over 90% at the training set size of 800. Thus, from the standpoint of predictive accuracy, both C4 and COGIN appear to scale similarly across problems in terms of number of examples required.

COGIN outperformed the decision tree results fairly consistently over all problems. Only on the simplest f6 multiplexor did COGIN fail to attain a higher absolute accuracy than the C4 decision tree at the largest training set size (98.1% vs 100%). However, given the probabilistic nature of COGIN's search, convergence to a correct solution on all runs was not really expected. In this case, COGIN did in fact achieve 100% accuracy

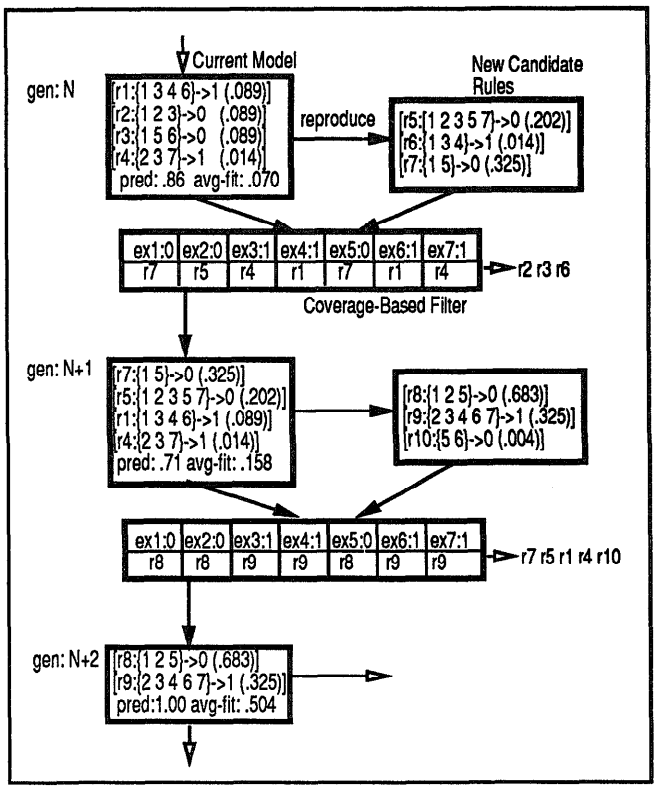

Figure 1: An example of COGIN's search process

consists of 4 rules r1, r2, r3, r4 which correctly predict 6 of the 7 examples (86%) in the training set and have an average rule fitness of .070. Three new rules are created on this cycle (r5, r6, r7), and the "allocation" of training examples within the filter (depicted) leaves no examples for rules r2, r3, or r6. These rules are discarded and the remaining rules (r7, r5, r1, r4) become the generation $N+1$ model. Notice that this model has the same number of rules of the generation N model, but actually a lower predictive accuracy. At the same time, the average rule fitness has increased, suggesting the emergence of better building blocks. After generating new rules and applying the filter, at Generation $N+2$ the new model is in fact seen to correctly classify all training set examples. While this example was created for illustrative purposes, this behavior is quite representative of the behavior indicated by the performance graph of actual system runs (2) discussed later in the paper.

In a recent paper[Greene & Smith 1992], we presented experimental results comparing the performance of COGIN with two other representative symbolic induction systems, CN2[Clark & Niblett 1989] and ID3[Quinlan 1986], that provided evidence of the utility of this approach. More specifically, COGIN demonstrated the leverage provided by genetic search across model building problems of varying complexity, with performance differential trends that suggest that this leverage increases with problem complexity. On a suite of 53 test data sets representing 2 and 4 class problems with increasingly complex generating functions and increasing levels of noise, COGIN performed significantly better than either compared system across all the experimental factors (number of classes, noise, form of data generating function).

At the same time, these results compared only the basic search techniques; no use of any pre or post processing techniques was made in running any of the three systems. Our goal in the next section is to examine the performance of COGIN's basic search process against the published performance of ID using well developed rule refinement procedures on a standardized multiplexor problem.

The Multiplexor Problem

The multiplexor family of problems is defined for $k = 1, \ldots$ as follows: an ordered list of $n = k + 2^k$ binary valued features is interpreted as a bit string of k address bits and 2^k data bits, where the binary encoding of the address bits indexes the particular data bit that is activated (turned on). The problem presents an interesting challenge for step-wise induction approaches, since address bits do not independently relate to classification outcome and the apparent discriminability of individual features is thus misleading.

Multiplexor problems have been the subject of prior comparisons of GA-Based and step-wise learning procedures. In [Wilson 1987], the ability to effectively learn multiplexors of size 6, 11, and 20 was reported using a classifier system approach. A subsequent study [Quinlan 1988] applied C4 (a decision-tree approach descended from ID3) to this same set of problems and, despite the potential difficulty raised above, achieved comparable levels of performance using much smaller numbers of training examples. Further research on this problem set within the incremental classifier system paradigm for GA-based learning (e.g. [Booker 1989, Liepins & Wang 1991] has yielded sizable reductions in the number of examples required, but still considerably more than required in [Quinlan 1988]. Arguments are made in [Liepins & Wang 1991] that, in fact, the classifier system paradigm can be expected to be inherently inefficient with respect to number of examples required in learning stationary Boolean concepts. In [DeJong & Spears 1991], a comparison of GABIL (a GA-based concept learner based alternatively on the Pitt paradigm of evolving complete classification models)

in 20 of the 25 runs that were averaged. Performance levels achieved by COGIN and the C4 decision tree on the 6 multiplexor problem at smaller training set sizes were fairly equivalent, with the exception of the smallest training set size of 50, where COGIN's absolute performance level exceeded even that of C4's post derived rules.

In the 11 multiplexor problem, COGIN's performance dominates that of the decision tree at all training set sizes and closely tracks the performance levels achieved by C4's post-derived rules. At the 400 training set size level, for example, COGIN achieved 100% accuracy on the 1000 unseen examples in 17 of the 25 averaged runs.

The progression of 20 multiplexor problems yielded the only cases where the performance of the C4 decision trees outperformed the models generated by COGIN. At the smallest training set sizes of 200 and 400, the performance differential is around 10%. At the training set size of 600, the differential narrows to 5% and COGIN's performance finally overtakes the decision tree's at a training set size of 800. Some accountability for this unexpected performance differential at lower training set levels might be attributable to characteristics of the specific training sets used by each system. But, with training set sizes representative of such low proportions of the instance space (e.g. .035% at the 400 level), it seems evident that entropy coupled with the additional lexicographic bias toward accurate rules does not provide sufficient search bias.

With respect to average size of the generated models (i.e., number of rules or decision tree nodes), a second interesting scaling relationship can be seen. In all three multiplexor problems, the size of the generated decision tree model increases monotonically with the size of the training set. In the case of COGIN, however, a different pattern is observed. The size of the model appears to increase with training set size only to a point, and then decreases fairly steadily as training set size is further increased. This pattern can be clearly seen in the 20 bit multiplexor problem. While it is difficult to extrapolate from the training set sizes tested, the observed pattern does suggest that the turning point occurs when a training set size large enough to provide sufficient grist for COGIN's search is reached. In the 11 multiplexor problem, the size of the model steadily decreases as increasingly larger training sets are provided. Thus, as the training set size is increased, the models generated by COGIN not only get more accurate but also more succinct. The sizes of the C4 post-derived rule sets are significantly smaller than COGIN's. However, here COGIN is at a distinct disadvantage, since a portion of the decision tree to rule set transformation involves selection of a default rule and a shift to a 1 class recognition model. This representational shift was not attempted within COGIN. Thus, for example in the 6 multiplexor problem, the minimal correct model for COGIN is 8 rules as opposed to 4 for C4.

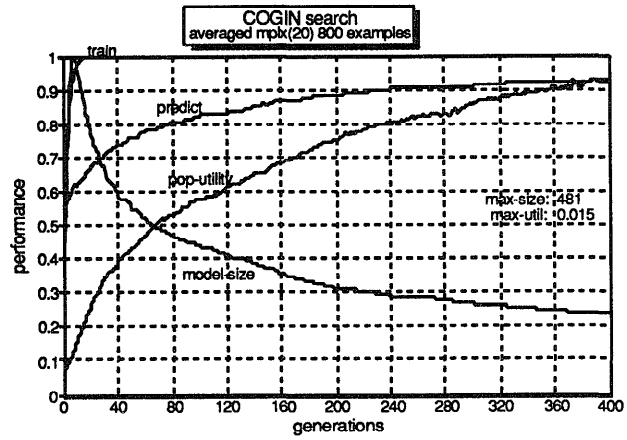

Figure 2: Performance curves for 800 example F20 problems

The dominant indicator of the computational complexity of COGIN's search process is the number of candidate rules evaluated. We indicated above that all COGIN experiments were run for a fixed number of generations (300 for F6; 600 for F11 and F20). Considering the average total number of individual rule evaluations performed over all F6 and F20 runs (6413 and 106,458 respectively), we find roughly a 9-fold computational increase in computational expense as in scaling from F6 to F20 (compared to the 8-fold increase reported for C4 in [Quinlan 1988]).

Finally, Figure 2 gives some insight the dynamics of COGIN's search. It tracks the averages of four values over time across one run on each of the five 800 example training sets for the 20 bit multiplexor problem: the % accuracy on the training set, the % accuracy on the 1000 example holdout set (of course unavailable to the system during the search), the average fitness of the rules in the current model, and the number of rules in the current model (the curves of the last two values are scaled to the maximum value seen). We can see that 100% accuracy on the training set is achieved very early on in the search, with the size of the current model generally increasing as well (a function of the dominant accuracy bias). At this point, the entropy bias takes over and focuses the search toward more general rules and correspondingly smaller models (rule sets).

Discussion

In this paper we examined the relative performance of a new GA-based framework for symbolic induction, embodied in the COGIN system, with an inductive system built around decision tree construction. While previous research has established the advantage of COGIN relative to basic stepwise search approaches, much work has gone into rectifying the limits of incremental approaches and it is these *total systems* which establish the performance standards for comparison. The multiplexor problem provides a search space which presents inherent difficulties for stepwise learning, and Quinlan's C4 demonstrates the large performance gains possible through post-processing. Nonetheless, the comparison presented showed the basic COGIN system to perform surprisingly well. COGIN was seen to generate models which were generally superior to pruned trees and often comparable to the converted rules. These results were achieved despite COGIN's reliance on a fairly minimal amount of search bias and no post processing. Our examination of generated models suggests opportunities for further improvement through post processing and this is one direction of our current research. Generally speaking, the results clearly demonstrate the viability of GAs in symbolic induction from examples contexts. They also complement the earlier results of [DeJong & Spears 1991] in refuting preliminary conclusions reported in [Quinlan 1988] about the comparative abilities of GA-based approaches.

Acknowledgments

This work was supported in part by the US ARMY Human Engineering Laboratory under contract DAAD05-90-R-0354, and the Robotics Institute.

References

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth Inc., 1984.

L. Booker. Triggered rule discovery in classifier systems. In D. Shaffer, editor, *Proceedings 3rd International Conference on Genetic Algorithms*, Washington, DC, June 1989. Morgan Kaufmann Publishers.

P Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4), March 1989.

K.A. DeJong and W.M. Spears. Learning concept classification rules using genetic algorithms. In *Proceedings 11th International Conference on Artificial Intelligence*, Syndey, Australia, August 1991. Morgan Kaufmann Publishers.

D.E. Goldberg. *Genetic Alorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing, 1989.

J.J. Grefenstette, C.L. Ramsey, and A.C. Schultz. Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5(4), October 1990.

D.P. Greene and S.F. Smith. Competition-based induction of decision models from examples. *Machine Learning*, (forthcoming).

R.C. Holte, L.E. Acker, and B.W. Porter. Concept learning and the problem of small disjuncts. In *Proceedings 11th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann Publishers, August 1989.

J.H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

J.H. Holland. *Escaping Brittleness: the Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-Based Systems*, volume 2. Tioga Press, Palo Alto, CA, 1986.

G.E. Liepins and L.A. Wang. Classifier system learning of boolean concepts. In *Proceedings 4th International Conference on Artificial Intelligence*, San Diego, CA, July 1991. Morgan Kaufmann Publishers.

J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3, 1989.

C.J. Matheus and L.A. Rendell. Constructive induction on decision trees. In *Proceedings 11th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann Publishers, August 1989.

J.R. Quinlan. The effect of noise on concept learning. In *Machine Learning: An Artificial Intelligence Approach*, volume II. Morgan Kaufmann Publishers, Palo Alto, CA, 1986.

J.R. Quinlan. Generating production rules from decision trees. In *Proceedings 10th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann Publishers, August 1987.

J.R. Quinlan. An empirical comparison of genetic and decision-tree classifiers. In *Proceedings 5th International Conference on Machine Learning.* Morgan Kaufmann Publishers, June 1988.

S.F. Smith. Flexible learning of problem solving heuristics via adaptive search. In *Proceedings 8th International Joint Conference on AI*, Karlsruhe, West Germany, August 1983.

S.W. Wilson. Classifier systems and the animate problem. *Machine Learning*, 2, 1987.