X-FCS: A Fuzzy Classifier System using Accuracy Based Fitness – First Results

Brian Carse
Intelligent Autonomous Systems Laboratory
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England, Bristol
Brian.Carse@uwe.ac.uk

Anthony G. Pipe
Intelligent Autonomous Systems Laboratory
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England, Bristol
Anthony.Pipe@uwe.ac.uk

Abstract

A fuzzy classifier system called X-FCS is proposed which employs accuracy-based fitness. Each classifier (rule) maintains an estimate of the payoff obtained when it fires and the accuracy of this prediction is used as the basis for selection under action of the genetic algorithm. The motivation behind this approach is that such a classifier system will be capable of generating compact, high-performance rule sets which are simultaneously general, accurate and co-adapted.

Keywords: Genetic Algorithms, Classifier Systems, Fuzzy Logic, Accuracy-Based Fitness.

1 Introduction

The fusion of rule-based representations and evolutionary algorithms has been, and continues to be, the focus of much attention as a basis for computational learning systems. The Learning Classifier System (LCS), devised by Holland [3] is an early example of a learning system that employs artificial evolution to evolve rule sets for problem solving. Following on from Holland's pioneering work, the LCS has been developed, refined and extended in many directions. A particularly significant direction has been to incorporate fuzzy sets and fuzzy inference into the LCS framework. Such Learning Fuzzy Classifier Systems are able to deal with problems where variables are real-valued, rule-bases are difficult to design by humans, and where linguistic interpretability is desirable.

2 Using the GA to Evolve Fuzzy Rule Bases

A major distinction among extant approaches in genetic optimisation of fuzzy system parameters is the way in which the GA is applied. With the so-called “Michigan” approach, the individual, as far as the GA is concerned, is a single rule or classifier. An alternative approach, called the “Pittsburgh” approach, maintains a population of rule-sets: each individual as far as the GA is concerned is a complete assembly of rules encoded on an appropriate genotype. Clearly the role of the GA in the two approaches is different as are the known difficulties: in Michigan-style systems a careful balance must be set between cooperation and competition between individual rules; in Pittsburgh-style systems reinforcement bandwidth is usually smaller and genetic crossover can be a cause of disruption. Indicative works using the Michigan approach include [1,5,7] and works using the Pittsburgh approach include [2,4]. Although the remainder of this paper focuses on the Michigan approach, but see our previous work, in [6], for a comparison of Pittsburgh and Michigan approaches in the same application domain as that considered here.

A commonly employed rule syntax used in fuzzy classifier systems is: IF (x_1 is A_1) AND (x_2 is A_2) .. AND (x_k is A_k) THEN (y_1 is B_1), (y_2 is B_2) .. (y_j is B_j)

where x_i are input variables, A_i are input membership functions, y_i are output variables and B_i are output membership functions. Such genotype codings are often supplemented with a “don’t care” label that indicates whether or not a particular part of the rule antecedent (or consequent) is inactive. Such “don’t cares” allow the representation of more general rules and the generation of more compact yet high-performance rule sets.

3 Two Key Classifier System Issues

3.1 Generalised Rules

In a discrete-valued classifier system, generalised rules are obtained by using '#' symbols ("don't cares') in the classifier syntax. The '#' symbol matches both '0' and '1' so, for example, the classifier condition 11## matches the input messages 1100, 1101, 1110 and 1111.

However, such a generalisation capability has been known for a long time to provide problems for discrete valued classifier systems and this also applies to the fuzzy case. The main problem is that of proliferation of "overgeneral" rules: rules which match many input states but whose outputs are only correct for a subset of input states and are incorrect for others. Despite being unreliable, such overgeneral rules can have more influence and better chances of survival (under action of the evolutionary algorithm) than other more specific and correct rules with which they compete.

3.2 Strength-Based versus Accuracy-Based Classifier Fitness

Traditional Michigan-style classifier systems have been "strength-based" in the sense that a classifier accrues strength during interaction with the environment (through rewards and/or penalties). This strength is then used for two purposes: resolving conflicts between simultaneously matched classifiers during learning episodes; and as the basis of fitness for the evolutionary algorithm. A number of problems arise from this dual use of classifier strength. These include:

1. The cooperation/competition problem. High-strength, potentially cooperative classifiers go on to compete under the action of the evolutionary algorithm.

2. Over-general rules with relatively high (but inconsistent) payoff can come to dominate the population.

3. In some environmental states, the maximum payoff achievable (by performing the best possible action for that state) may be relatively low. Although a classifier might be the best that can exist for that state, it can be eradicated from the population by other classifiers that achieve higher rewards in other states. This results in gaps in the system's "covering map".

In [8] a completely different approach is taken in which a classifier's fitness, from the point of view of the evolutionary algorithm, is based on its "accuracy" i.e. how well a classifier predicts payoff whenever it fires. Such an accuracy based approach offers a number of advantages. Firstly, it can distinguish between accurate and overgeneral classifiers: an overgeneral classifier will have relatively low accuracy since payoff will vary according to the input states covered by the classifier. Indeed, it has been shown that the accuracy-based approach can lead to evolution of optimally general classifiers. Additionally it can maintain both consistently correct and consistently incorrect classifiers which allows learning of a complete "covering map". A potential drawback of the accuracy-based approach is that it is likely to require larger populations of classifiers.

4 An Accuracy-Based Fuzzy Classifier System (X-FCS)

4.1 Classifier Syntax

In this initial implementation of X-FCS, each classifier encodes a string of integers representing a fuzzy set membership function for each system input and output variable. Each integer may take a value in the range $[0,N_{fs}]$ where N_{fs} is the number of fuzzy sets in the appropriate universe of discourse. A value of 0 in the encoding of an input fuzzy set represents the “don’t care” symbol. The complete classifier system rule base contains a variable number of such fuzzy rules that are randomly initialised.

4.2 Classifier Execution Cycle

On each classifier execution cycle, an input vector is read in from the environment. If this vector is not matched by any rules, a cover operator is applied which generates a random matching classifier that is inserted into the population. A *match set* containing all classifiers that match to a degree greater than zero is created. This match set is further decomposed into subpopulations where each classifier in each subpopulation has the same antecedent (including don’t cares). A classifier is selected from each separate subpopulation and these classifiers form the action set. These classifiers are fired and the outputs aggregated and defuzzified to provide the crisp classifier system output. Any environmental reward obtained is then used to update the fired classifiers’ strengths, predictions and accuracies. Note that an overgeneral classifier will fire frequently with different payoffs and the predicted payoff learned for that classifier will not converge; such a classifier will have low accuracy and therefore low reproductive fitness when the GA is applied.

4.3 Evolutionary Operators

The genetic algorithm is applied to individual subpopulations when the majority of active classifiers have fired at least a certain number of times (a system parameter) and payoff predictions for accurate classifiers have stabilised. Classifiers for reproduction are chosen using roulette-wheel selection on the subpopulations. Two-point crossover is applied to the chosen parent classifier strings. Two mutation operators are applied (with small probability) to offspring: a “creep” mutation operator which adds or subtracts one from a randomly chosen membership function; and a more explorative mutation operator which replaces a randomly chosen membership function with a random integer in the range $[0,N_{fs}]$.

5 Experimental Evaluation

Initial experimental results have been carried out comparing accuracy-based fitness and strength-based fitness on a mobile robotics problem. This “stimulus-response” obstacle avoidance learning task requires no internal linkage between fuzzy rules, and environmental reward is immediate. The robot has five ultrasonic distance sensors, as shown in figure 1.

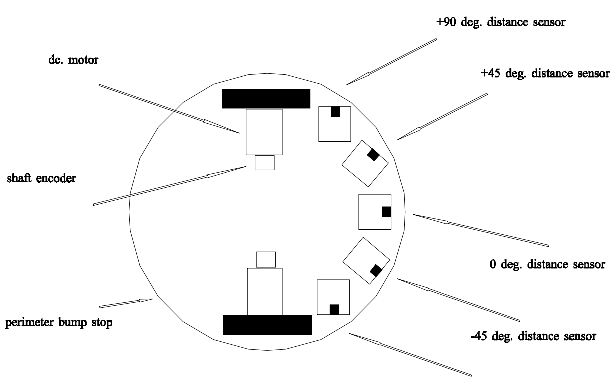

Figure 1: Robot used in experiments

The fuzzy system has these five inputs, and a single “steering angle” output that is used by a lower level controller to drive the twin-wheel differential drive at a constant forward speed. The fuzzy controller’s input domains each have three membership functions, whilst the single output domain has 7, distributed as shown in figure 2. Details of parameter values used for these experiments can be found at the following website:

(http://www.ias.uwe.ac.uk/~t-pipe/index.html)

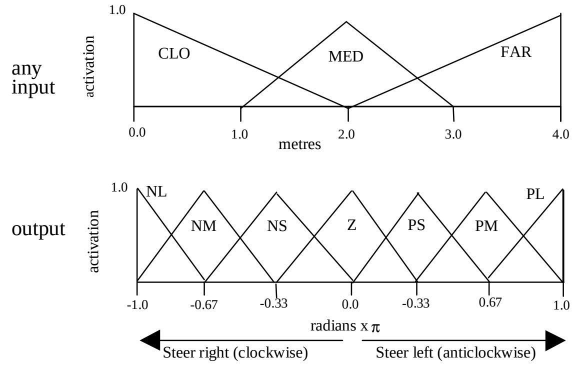

Figure 2: Membership functions

“Strength” and “accuracy” regimes were tested, but in both cases it was necessary to include a factor that helped to prevent stationary or closely-coupled cyclic behaviour from developing during the evolutionary process. Environmental reward was therefore subjected to a multiplying factor related to the distance travelled from the start position at that time in a trial, making the same rule firing at some strength later in a trial “worth more” than an earlier firing at the same strength.

In the “strength-based” scheme, fitness was dependent on a filtered average of the strength of each rule firing in the fuzzy system over the duration of the trial. In the “accuracy-based” scheme fitness was derived through a similar process, but where the reciprocal of the variance from the average reward was used as the fitness for each rule.

The 10-metre by 10-metre maze is shown in figure 3, along with an example of a robot trajectory, starting at the top of the figure, generated by a high fitness controller.

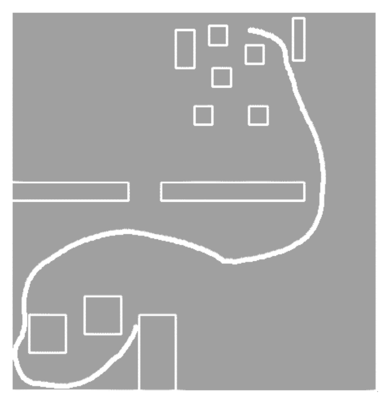

Figure 3: Maze & robot trajectory

The trajectory shown in figure 3 was generated by the “strength-based” fitness regime. The “accuracy-based” scheme that was implemented for these experiments was not so successful, as illustrated by the plots in figure 4, showing distances from the start position attained by the two schemes over successive generations. This figure shows the best controller at each generation and the average performance of all controllers for each fitness scheme. For the experiments illustrated by figure 4, the “accuracy-based” population was 7-times larger than the “strength-based” population. This was done to allow for the fact that the former learning scheme is attempting to acquire high fitness rules across the

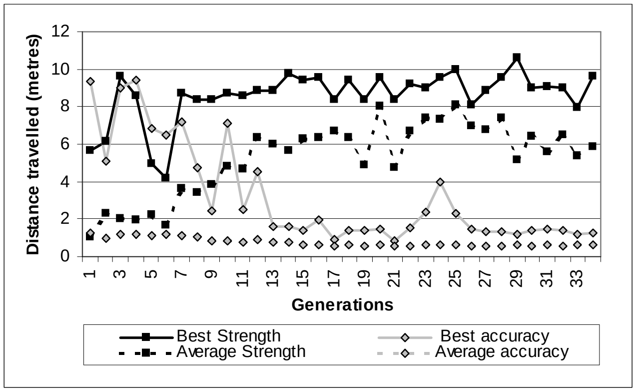

Figure 4: strength- and accuracy-based comparison

entire input-output mapping, not just the optimal mapping. Despite this, one can easily see the high-strength population members being lost from the “accuracy-based” scheme. This trend is contrasted strongly by the “strength-based” plots.

6 Conclusions and Further Work

The “strength-based” fitness approach achieved better convergence velocity on high quality solutions than any attained during our previous work in this area [6]. This was achieved by niching classifiers into GA subpopulations, thus removing problems of competition between co-operating classifiers.

Further experiments are required to fully understand the reasons for low performance in the “accuracy-based” approach. However, we can propose a plausible initial hypothesis. The fitness landscape is dynamic, because of the increasing value that is attached to actions as a maze trial proceeds. Any repeated non-zero strength action will incur a prediction error proportional to this factor in the reward scheme. By contrast, a zero-strength rule can be extremely accurate here. Examination of the population members at the end of an “accuracy-based” trial, does indeed reveal a predominance of high fitness very low-strength rules. Of course, this is a feature of an accuracy-based scheme, i.e. the Classifier System is acquiring rules that span the solution space from good to bad rules, the requirement is only that those rules are *accurate* in their predictions. This form of learning has powerful potential for robustness in acquiring rules that make accurate predictions in all circumstances *not* just the optimal ones This is very important for a number of problem domains, including mobile robotics. We believe, this makes “accuracy-based” fitness worthy of further work beyond this preliminary study.

References

[1] Bonarini A. (2000). An Introduction to Learning Fuzzy Classifier Systems. In P.L. Lanzi, W. Stolzmann and S.W. Wilson (Eds.), Learning Classifier Systems- from Foundations to Applications, Lecture Notes in Artificial Intelligence, pages 83-104. Springer Verlag Berlin Heidelberg, Germany.

[2] Carse B., Fogarty T.C. and Munro A. (1996). Evolving Fuzzy Rule-based Controllers using Genetic Algorithms. *Fuzzy Sets and Systems* 80(3), pages 273-293.

[3] Holland J.H. (1976). Adaptation. In R. Rosen and F.M.Snell (Eds.), Progress in Theoretical Biology, New York:Plenum.

[4] Hwang W. and Thompson W. (1994). Design of Fuzzy Logic Controllers using Genetic Algorithms. In *Procs. of the 3rd IEEE International Conference on Fuzzy Systems*, pages 1383-1388, Piscataway, NJ: IEEE Computer Press.

[5] Parodi A. and Bonelli P. (1993). A New approach to fuzzy classifier systems. In *Procs. of 5th International Conference on Genetic Algorithms* pages 223-230. San Mateo, CA:Morgan Kaufman.

[6] Pipe A.G., Carse B. (2000). Autonomous Acquisition of Fuzzy Rules for Mobile Robot Control: First Results from two Evolutionary Computation Approaches. In *Procs. of Genetic & Evolutionary Comp. GECCO 2000*, pages 849-856.

[7] Valenzuela-Rendon M. (1991). The Fuzzy Classifier System: a Classifier System for Continuously Varying Variables. In *Proceedings of the Fourth Int. Conference on Genetic Algorithms*, pages 346-353, San Mateo, CA:Morgan Kaufman.

[8] Wilson S. W. (1995). Classifier Fitness based on Accuracy. *Evolutionary Computing* 3(2):149-175.