A new approach to encoding actions in classifier systems

Lashon B. Booker

The MITRE Corporation
1820 Dolley Madison Blvd
McLean, VA 22102-3481, USA
booker@mitre.org

Extended Abstract

The classifier system framework is a general-purpose approach to learning and representation designed to exhibit non-brittle behavior in complex, continually varying environments. Broadly speaking, classifier systems are expected to avoid brittle behavior because they implement processes that build and refine models of the environment. One of the most important of these processes is categorization. As Holland [5] has pointed out (p. 598) "Categorization is the system's major weapon for combating the environment's perpetual novelty. The system must readily generate categories for input messages, and it must be able to generate categories relevant to its internal processes". Research in classifier systems has focused almost exclusively on finding generalizations for input messages. However, generalizations of actions will also be required in order to build effective models of the environment. This paper introduces a new encoding for actions in classifier rules that lends itself to representing abstract actions.

What categorizations of actions are desirable? Two distinct kinds of generalizations appear to be useful [4]: abstractions that generalize over the possible outcomes of an action, and aggregations that group related actions together. Both kinds of generalizations are important for classifier systems, and some previous research has offered preliminary ideas about how to represent them (e.g., the expectation part of rules and action chunks in ACS [8] and the fuzzy combination of action effects in fuzzy classifier systems [1]). One fundamental issue that has not been addressed, though, is that actions are always represented as primitive symbols having no constituent parts. This is problematic in a classifier system that depends on a genetic algorithm to discover generalizations by manipulating building blocks.

What is needed is an encoding for actions that uses effector settings as the parts of an action, while at the same time permits useful generalizations of some kind. Mechanisms prevalent in natural systems point the way to a solution. Regarding actions themselves, pairs of interacting effectors can provide a powerful set of building blocks for constructing more sophisticated behaviors [2]. As for encodings, the genotype of a heterozygote can be viewed as a reservoir of variability [3] that in some sense is an abstraction of the specific phenotypic traits that can be produced from that genotype by breeding. The traits are the physical manifestations (or outcomes) of the actions of the genes. The remainder of this paper describes how these two ideas can be combined in the classifier systems framework to provide a new approach to encoding actions.

Consider some trait controlled by a single gene, and assume that neither of the two alleles A-a for that gene is dominant. By convention, the allele denoted by the capital letter is the one responsible for increased manifestation of the trait in question. Under these conditions, the diploid heterozygotes Aa will produce a phenotype that is intermediate between the effects seen in the homozygotes AA and aa. If the trait is height, for example, the homozygotes will be tall or short while the heterozygotes will be "average". When more than one gene controls a trait, and those genes have equal and additive effects on the phenotype (assuming again that there is no dominance), then more variation will be observed in the range of phenotypes. In the case of two genes controlling height, for example, we have phenotypes (genotypes) "tall" (AABB), "short" (aabb), "moderately tall" (AaBB, AABb), "moderately short" (aaBb,Aabb), and "average" (AAbb,aaBB,AaBb). If there are three genes of equal effect there will be seven distinct phenotypes. In general, with n genes there will be $2n+1$ distinct phenotypes. Note that underlying the phenotypes is a graded series of genotypes and, except for the two phenotypic extremes, each phenotype can be produced by a number of genotypes.

The notion of several genes making equal, additive contributions to an observable outcome suggests an action representation for classifier rules that I will call the *paired effector encoding*. Assume for the moment that we have two actions 0 and 1 controlled by a pair of effectors. Each effector "votes" for which action to take and the action receiving the most votes is the one selected. There are several possibilities for handling tie votes. If the actions are mutually exclusive then a tie could lead to a random choice between the actions. If it makes sense to "blend" the actions into some resultant behavior — for example, equal tendencies to turn left and right result in a movement straight ahead — then ties could produce that resultant. There are undoubtedly other ways to handle ties given different assumptions about how the actions interact. Regardless of the semantics, though, the action choice can be encoded using a ternary alphabet $\{0, 1, ?\}$ to designate the outcomes 0, 1, and "tie". Finer gradations in outcome can be represented by assuming the action choice is controlled by the combined influence of several pairs of effectors. If there are n effector pairs then $2n+1$ distinct outcomes can be represented in a string of length n. For example, using $n = 2$ gives the following five outcomes (assuming ties mean a random choice): 100% action 1 (11), 100% action 0 (00), 75% action 1 (1?,?1), 75% action 0 (0?,?0), 50% action 1 (10,01,??). Note that these outcomes are directly analogous to the range of phenotypes discussed earlier, where the ternary alphabet corresponds to the distinguishable single-locus phenotypes AA, aa, and Aa.

There are several potential advantages to using this paired-effector action encoding. First, it provides a graded representation of the space of possible actions in a manner that may facilitate genetic search. Assuming, for example, that strings having a nondecreasing number of "votes" for the best action will receive correspondingly nondecreasing fitness evaluations (i.e., the more often the best action wins, the higher the fitness), the search problem is basically the GA-easy counting 1's problem [9]. Second, the genetic variability provided by multiple genotypes per phenotype should help maintain useful diversity in the action gene pool without using disruptive mutation rates. Third, since the encoding can specify more than one action, each rule can have a variety of effects. This property may help "smooth" the consequences of inserting or deleting rules [7]. Finally, by allowing the specification of stochastic action choices that can evolve into deterministic choices, this representation supports the implementation of stochastic policies which are known to be useful in solving partially observable Markovian decision processes [6].

Experiments are currently being conducted to investigate the advantages and disadvantages of the paired-effector encoding.

References

1. Andrea Bonarini, Claudio Bonacina, and Matteo Matteucci. Fuzzy and crisp representations of real-valued input for learning classifier systems. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Learning Classifier Systems: From Foundations to Applications*, volume 1813 of *LNAI*, pages 107–124, Berlin, 2000. Springer-Verlag.
2. Valentino Braitenberg. *Vehicles: Experiments in synthetic psychology.* Cambridge, MA, 1984. The MIT Press.
3. C. Darlington and K. Mather *The Elements of Genetics.* London, 1950. Allen and Unwin LTD.
4. Peter Haddawy and AnHai Doan. Abstracting probabilistic actions. In R. Lopez de Mantaras and D. Poole, editors, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 270–277, Seattle, WA, 1994. Morgan Kaufmann.
5. John H. Holland. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell, editors, *Machine learning: An artificial intelligence approach*, volume II, pages 593–623, Los Altos, CA, 1986. Morgan Kaufmann.
6. Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In W. Cohen and H. Hirsh, editors, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 284–292, New Brunswick, NJ, 1994. Morgan Kaufmann.
7. Robert E. Smith, B. A. Dike, B. Ravichandran, A. El-Fallah, and R. K. Mehra. The fighter aircraft LCS: A case of different LCS goals and techniques. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Learning Classifier Systems: From Foundations to Applications*, volume 1813 of *LNAI*, pages 283–300, Berlin, 2000. Springer-Verlag.
8. Wolfgang Stolzmann. An introduction to Anticipatory Classifier Systems. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Learning Classifier Systems: From Foundations to Applications*, volume 1813 of *LNAI*, pages 175–194, Berlin, 2000. Springer-Verlag.

9. Michael D. Vose and Gunar E. Liepins. Schema disruption. In R. Belew and L. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA91)*, pages 237–242, San Diego, CA, 1991. Morgan Kaufmann.