Machine Learning, 5, 351–353 (1990)

Introduction

This second special issue on Genetic Algorithms (the first appeared as Volume 3, numbers 2–3) was motivated by the quality and variety of papers presented at the Genetic Algorithms Conference in June of 1989. Authors of papers which focused on issues and problems of particular interest to the machine learning community were encouraged to submit expanded versions to be considered for inclusion in this special issue. Of the six papers selected, four appear in this issue. Because of time and space constraints, the remaining two will appear independently in later issues.

There have been several attempts to provide taxonomies of the field of machine learning on the basis of methods employed, problem domains, internal knowledge representations, etc. Genetic algorithms (GAs) have never fit comfortably into any of them. The reason for this is that genetic algorithms are fairly general-purpose adaptive search procedures. As such they (like other search algorithms such as *depth-first tree traversal with alpha-beta cutoffs*) make few, if any, *a priori* assumptions about the problem domains to which they are applied. Consequently, the range of applications to which such general techniques are applied is limited only by the imagination and ingenuity of the system designer.

In the case of genetic algorithms, the applications span a wide variety of disciplines and problem classifications including classical function optimization problems, NP-complete problems (such as Boolean satisfiability, job-shop scheduling, and VLSI routing problems), image registration problems, problems involving adapting both the structure and weights of neural networks, concept learning tasks, learning reactive plans, and many other tasks. The proceedings of the three GA Conferences (Grefenstette, 1985; Grefenstette, 1987; Schaffer, 1989) provide an excellent snapshop of this diversity. The key point in deciding whether or not to use genetic algorithms for a particular problem centers around the question: what is the space to be searched? If that space is well-understood and contains structure that can be exploited by special-purpose search techniques, the use of genetic algorithms is generally computationally less efficient. If the space to be searched is not so well understood and relatively unstructured, and if an effective GA representation of that space can be developed, then GAs provide a surprisingly powerful search heuristic for large, complex spaces (Holland, 1985; De Jong, 1988; Goldberg, 1989).

The most straightforward kinds of machine learning problems which meet these criteria are parameter tuning problems which map nicely into fixed-length binary strings, one of the most well-understood GA representations. Although parameter tuning seems somewhat mundane in contrast to *concept learning*, there are a surprising number of non-trivial learning problems which fit into this category including Samuel's checker player and weight adjustment in neural networks. Dave Montana's article in this issue provides an excellent real-world example of this type of GA-based learning. He describes an operational system for improving and adapting the performance of sonar classification rules over time. He uses more traditional knowledge engineering techniques to derive sonar classification rules from human experts. However, the rules themselves involve a large number of "guestimated"

thresholds which are frequently interacting and suboptimal. GAs are used to tune these thresholds to current and changing operational sonar environments.

John Holland provided a more complex paradigm for applying GAs to machine learning problems with the development of his classifier systems (Holland, 1986). Classifier systems are capable of learning collections of situation-action rules for dealing with complex and unknown environments. This is accomplished by having rules compete for space in a limited-size short-term rule memory. Rules gain competitive strength by participating in chains of rule firings which lead to a reward from the environment. GAs are used to replace weaker rules by offspring of stronger rules. Classifier systems have been successfully applied to a variety of rule-learning tasks including sequence prediction, gas pipeline control, traversing difficult mazes, and learning Boolean concepts.

Two of the papers in this issue describe important extensions to Holland's classifier systems. Harry Zhou has incorporated a long-term memory component which permits the storing and retrieving of task-specific rule sets. This significantly improves the robustness of classifier systems in situations in which multiple tasks must be learned over long periods of time. Dave Goldberg addresses the difficult issue of designing learning systems which are robust and useful in environments which contain uncertainty and may be changing while learning is taking place. By making a subtle, but well-motivated, change to a classifier system's internal bidding scheme, he is able to show that classifier systems can learn effectively even in such harsh conditions.

Both of the types of GA-based learning just discussed (parameter tuning and classifier systems) typically represent the parameters and rules internally as fixed-length binary strings. This has led to the fairly widely held view in the machine learning community that GAs are sub-symbolic learning mechanisms. To reiterate, however, the point made earlier: GAs are fairly general purpose adaptive search procedures. There is no *a priori* reason why they could not be used to effectively search symbolic spaces as well. The theoretical analysis of GAs suggests that their power comes from representations and operators which permit the formation of useful building blocks (partial solutions) from which complete solutions can be constructed. Historically, this has been easiest to analyze and achieve with binary fixed-length strings. However, we are now beginning to understand and develop GA-based learning systems for symbolic domains such as evolving Lisp programs, traditional concept learning, and learning high-level IF-THEN rules sets (Grefenstette, 1987; Schaffer, 1989). The paper in this issue by Grefenstette, Ramsey, and Schultz is an excellent example of this kind of GA-based learning. They describe their Samuel system, a general architecture for using GAs to learn sequential decision rules via simulation, and provide an interesting and impressive set of empirical results on an evasive manuevering problem.

It has been a pleasure to pull together this special GA issue. I feel that it provides a good picture of the type and variety of GA applications to machine learning problems. Readers interested in staying abreast of the latest GA activities should consider subscribing to the electronic bulletin board GA-LIST. This can be accomplished by sending email to GA-LIST-REQUEST@aic.nrl.navy.mil.

Kenneth De Jong
Computer Science Department and AI Center
George Mason University, Fairfax, VA 22030
(KDEJONG@AIC.GMU.EDU)

References

De Jong, K. (1988). Learning with genetic algorithms: An overview. *Machine Learning, 3,* 121–138.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley Publishing.

Grefenstette, J. (1985). *Proceedings of the First International Conference on Genetic Algorithms and their Applications*. Pittsburgh, PA: Lawrence Erlbaum Publishers.

Grefenstette, J. (1987). *Proceedings of the Second International Conference on Genetic Algorithms and their Applications*. Cambridge, MA: Lawrence Erlbaum Publishers.

Holland, J.H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Holland, J.H. (1986). Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.). *Machine learning: An artificial intelligence approach* (Vol. 2). San Mateo, CA: Morgan Kaufman Publishing.

Schaffer, D. (1989). *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishing.