Machine Learning 3: 95–99, 1988


# GUEST EDITORIAL

# Genetic Algorithms and Machine Learning

## Metaphors for learning

There is no *a priori* reason why machine learning must borrow from nature. A field could exist, complete with well-defined algorithms, data structures, and theories of learning, without once referring to organisms, cognitive or genetic structures, and psychological or evolutionary theories. Yet at the end of the day, with the position papers written, the computers plugged in, and the programs debugged, a learning edifice devoid of natural metaphor would lack something. It would ignore the fact that all these creations have become possible only after three billion years of evolution on this planet. It would miss the point that the very ideas of adaptation and learning are concepts invented by the most recent representatives of the species *Homo sapiens* from the careful observation of themselves and life around them. It would miss the point that natural examples of learning and adaptation are treasure troves of robust procedures and structures.

Fortunately, the field of machine learning does rely upon nature's bounty for both inspiration and mechanism. Many machine learning systems now borrow heavily from current thinking in cognitive science, and rekindled interest in neural networks and connectionism is evidence of serious mechanistic and philosophical currents running through the field. Another area where natural example has been tapped is in work on *genetic algorithms* (GAs) and genetics-based machine learning. Rooted in the early cybernetics movement (Holland, 1962), progress has been made in both theory (Holland, 1975; Holland, Holyoak, Nisbett, & Thagard, 1986) and application (Goldberg, 1989; Grefenstette, 1985, 1987) to the point where genetics-based systems are finding their way into everyday commercial use (Davis & Coombs, 1987; Fourman, 1985).

## Genetic algorithms and classifier systems

This special double issue of *Machine Learning* is devoted to papers concerning genetic algorithms and genetics-based learning systems. Simply stated, genetic algorithms are probabilistic search procedures designed to work on large spaces involving states that can be represented by strings. These methods are inherently parallel, using a distributed *set* of samples from the space (a population of strings) to generate a new set of samples. They also exhibit a more subtle *implicit parallelism.* Roughly, in processing a population of $m$ strings, a genetic algorithm implicitly evaluates substantially more than $m^3$ component substrings. It then automatically biases future populations to exploit the above average components as *building blocks* from which to construct structures that will exploit regularities in the environment (problem space). Section 3 of the paper by Fitzpatrick and Grefenstette gives a clear discussion of this property. The theorem that establishes this speedup and its precursors – the *schema* theorems – illustrate the central role of theory in the development of genetic algorithms. Learning programs designed to exploit this building block property gain a substantial advantage in complex spaces where they must discover both the "rules of the game" and the strategies for playing that "game."

Although there are a number of different types of genetics-based machine learning systems, in this issue we concentrate on *classifier systems* and their derivatives. Classifier systems are parallel production systems that have been designed to exploit the implicit parallelism of genetic algorithms. All interactions are via standardized messages, so that conditions are simply defined in terms of the messages they accept and actions are defined in terms of the messages they send. The resulting systems are computationally complete, and the simple syntax makes it easy for a genetic algorithm to discover building blocks appropriate for the construction of new candidate rules. Because classifier systems rely on competition to resolve conflicts, they need no algorithms for determining the global consistency of a set of rules. As a consequence, new rules can be inserted in an existing system, as trials or hypotheses, without disturbing established capacities. This *gracefulness* makes it possible for the system to operate incrementally, testing new structures and hypotheses while steadily improving its performance.

## Arguments for the evolutionary metaphor

These attractive properties of genetics-based systems – explicit parallelism, implicit parallelism, and gracefulness – are explored more fully in the papers that follow. However, before proceeding further we must answer an important question. Of the two natural archetypes of learning available to us – the brain and evolution – why have genetic algorithm researchers knowingly adopted the "wrong" metaphor? One reason is expedience. The processes of natural evolution and natural genetics have been illuminated by a century of enormous progress in biology and molecular biology. In contrast, the brain, though yielding some of its secrets, remains largely an opaque gray box; we can only guess at many of the fundamental mechanisms contained therein.

Of course, simple expedience is not the best reason for adopting a particular course of action, and at first glance, it is not at all obvious why learning in natural or artificial minds should be anything like the adaptation that has occurred in evolution. Yet there is an appealing symmetry in the notion that the mechanisms of natural learning may resemble the processes that created the species possessing those learning processes. Furthermore, the idea that the mind is subject to the same competitive-cooperative pressures as evolutionary systems has achieved some currency outside of GA circles (Bateson, 1972; Edelman, 1987; Minsky, 1986).

Despite these suggestions, genetic algorithms and genetics-based machine learning have often been attacked on the grounds that natural evolution is simply too slow to accomplish anything useful in an artificial learning system; three billion years is longer than most people care to wait for a solution to a problem. However, this slowness argument ignores the obvious differences in time scale between natural systems and artificial systems. A more fundamental fault is that this argument ignores the robust complexity that evolution has achieved in its three billion years of operation. The 'genetic programs' of even the simplest living organisms are more complex than the most intricate human designs.

Waddington (1967) presents more sophisticated probabilistic arguments that actual evolutionary processes have achieved a complexity in existing species that is incommensurate with an evolutionary process using only selection and mutation. Although such arguments were originally meant to challenge evolutionary theory, genetic algorithmists see no such challenge. Instead, the high speed-to-complexity level observed in nature lends support to the notion that reproduction, recombination, and the processing of building blocks result in the *rapid* development of appropriate complexity. Moreover, this speed is not purchased at the cost of generality. The mechanisms of genetics and genetic algorithms permit relative efficiency across a broad range of problems.

## Contents of the special issue

This robust combination of breadth and efficiency is a recurring theme in work on genetic algorithms, and any collection of papers on the topic is likely to cover a broad range. The current set of papers has been selected to give a representative view of the major lines of research involving genetic algorithms. The first paper, by Fitzpatrick and Grefenstette, discusses the theory and application of a genetic algorithm in a difficult, noisy search domain – medical image registration. Next De Jong provides an overview and careful discussion of alternative approaches to machine learning using genetic algorithms. The third paper, by Robertson and Riolo, explores the problem of "scaling up" when one implements 8000 rule classifier systems on a massively parallel machine. After this, Booker discusses experiments with a simulated roving automaton, bridging the gap between theories of animal learning and machine learning. In the fifth paper, Belew and Forrest compare symbolic and subsymbolic approaches to machine learning, using a classifier-system implementation of KL-ONE as their starting point. The final paper, by Grefenstette, experimentally compares various methods for credit assignment in these highly parallel systems.

The abstracts provide an effective annotated table of contents that we will not try to duplicate here. However, it is worth extracting a few comments from the papers concerning the kinds of issues typical of research on genetic algorithms:

> Genetic algorithms search by allocating effort to regions of the search space based on an estimate of the relative performance of competing regions. [In complex domains] one expects perpetual novelty to be a characteristic feature ... In these cases, traditional search techniques ... are likely to be misled [and] genetic algorithms may be the search technique of choice for machine learning systems ... [Fitzpatrick and Grefenstette]

> If very little background knowledge is available and the problem environment provides a natural measure for the quality of outcomes, it is appropriate to view the problem of learning as a search for high-performance structures ... [Grefenstette]

> ... the syntactic and semantic complexity of traditional languages makes it difficult to develop [structure-modifying] operators ... that preserve ... the syntactic integrity ... of the programs being manipulated ... An obvious next step [is to] focus on less traditional languages with simpler syntax and semantics [that can be subjected to] genetic operators. [De Jong]

> [It is important to provide] low-level learning systems that [support] representations at the symbolic level [and that allow] integration of programmed and learned knowledge ... [Belew and Forrest]

> ... sequences of coupled classifiers, ... [or] *classifier chains*, are necessary so that classifiers can be used to implement arbitrary networks and to perform a variety of computations. ... the system must not only be able to *discover* (create) a set of classifiers to solve the problem, it must also be able to *maintain* such a set once it has been discovered. [Robertson and Riolo]

> [An internal] model is used to direct behavior, and learning is triggered whenever the model proves to be an inadequate basis for generating behavior in a given situation. This means that overt external rewards are not necessarily the only or the most useful source of feedback for inductive change. [Booker]

Although the papers in this special issue are representative, they can only suggest the breadth of current activity. The proceedings from two conferences on genetic algorithms (Grefenstette, 1985, 1987), held at Carnegie Mellon University in 1985 and at Massachusetts Institute of Technology in 1987, contain papers ranging from VLSI layout compaction to problem-directed generation of LISP code. The diversity and level of this activity are the signposts of a journey that has just begun. Along the way, researchers have already learned that evolutionary processes are not "slow," and that discovery and recombination of building blocks, allied with speedup provided by implicit parallelism, provides a powerful tool for learning in complex domains. As the journey continues, we are confident that an approach based on the abstraction of natural example, combined with careful theoretical and computational investigation, will continue to chart useful territory on the landscape of machine learning.

David E. Goldberg
University of Alabama, Tuscaloosa
DGOLDBER@UA1VM.BITNET

John H. Holland
University of Michigan, Ann Arbor
JHH@UM.CC.UMICH.EDU

## References

Bateson, G. (1972). *Steps to an ecology of mind.* New York: Ballantine.

Davis, L., & Coombs, S. (1987). Genetic algorithms and communication link speed design: Theoretical considerations. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (pp. 252–256). Cambridge, MA: Lawrence Erlbaum.

Edelman, G. M. (1987). *Neural Darwinism: The theory of neuronal group selection.* New York: Basic Books.

Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications* (pp. 141–152). Pittsburgh, PA: Lawrence Erlbaum.

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

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

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

Holland, J. H. (1962). Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery, 3*, 297–314.

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

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction: Processes of inference, learning, and discovery.* Cambridge, MA: MIT Press.

Minsky, M. (1986). *The society of mind.* New York: Simon and Schuster.

Waddington, C. H. (1967). Summary discussion. In P. S. Moorhead & M. M Kaplan (Eds.), *Mathematical challenges to the neo-Darwinian interpretation of evolution.* Philadelphia, PA: Wistar Institute Press.