Introduction: Paradigms for Machine Learning

Jaime G. Carbonell
School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A.

1. Historical Perspectives

Machine learning (ML) played a central role in artificial intelligence at its very beginning. Although the AI limelight has wandered away from machine learning in the advent of other significant developments, such as problem solving, theorem proving, planning, natural language processing, robotics, and expert systems, ML has returned cast in multiple guises, playing increasingly more significant roles. For instance, early work in linear perceptrons faded away in light of theoretical limitations, but resurged this decade with much fanfare as connectionist networks with hidden units able to compute and learn nonlinear functions. In the interim, many symbolic machine learning paradigms flourished, and several have evolved into powerful computational methods, including inductive concept acquisition, classifier systems, and explanation-based learning. Today, there are many active research projects spanning the gamut of machine learning methods, several focusing on the theory of learning and others on improving problem solving performance in complex domains. In the 1980s, the field of machine learning has re-emerged one of the major areas of artificial intelligence, with an annual ML conference, an established 1,000-subscriber journal, dozens of books, and ample representation in all major AI conferences.

Perhaps the tenacity of ML researchers in light of the undisputed difficulty of their ultimate objectives, and in light of early disappointments, is best explained by the very nature of the learning process. The ability to learn, to adapt, to modify behavior is an inalienable component of human intelligence. How can we build truly artificially intelligent machines that are not capable of self-improvement? Can an expert system be labeled "intelligent," any more than the *Encyclopedia Britanica* be labeled intelligent, merely because it contains useful knowledge in quantity? An underlying conviction of many ML researchers is that learning is a prerequisite to any form of true intelligence—

therefore it must be investigated in depth, no matter how formidable the challenge. Philosophical considerations aside, machine learning, like knowledge representation and reasoning, cuts across all problem areas of AI: problem solving, theorem proving, analogical and nonmonotonic reasoning, natural language processing, speech recognition, vision, robotics, planning, game playing, pattern recognition, expert systems, and so on. In principle, progress in ML can be leveraged in all these areas; it is truly at the core of artificial intelligence.

Recently, machine learning research has begun to pay off in various ways: solid theoretical foundations are being established; machine learning methods are being successfully integrated with powerful performance systems; and practical applications based on the more established techniques have already made their presence felt. Recent successes in machine learning include decision tree induction applied to industrial process control (based on Quinlan's ID3 [14] and its successors), the integration of explanation-based learning into general knowledge-intensive reasoning systems (such as SOAR [9], PRODIGY [11] and THEO), and extended forms of neural network learning to produce phonemic-level speech recognition at an accuracy surpassing conventional methods (such as hidden Markoff models) in modular time delay neural networks.

To date one can identify four major ML paradigms and multiple sub-paradigms under active investigation: inductive learning (e.g., acquiring concepts from sets of positive and negative examples), analytic learning (e.g., explanation-based learning and certain forms of analogical and case-based learning methods), genetic algorithms (e.g., classifier systems) [7]), and connectionist learning methods (e.g., nonrecurrent "*backprop*" hidden layer neural networks). These machine learning paradigms emerged from quite different scientific roots, employ different computational methods, and often rely on subtly different ways of evaluating success, although all share the common goal of building machines that can learn in significant ways for a wide variety of task domains. In all cases, learning can be defined operationally to mean the ability to perform new tasks that could not be performed before or perform old tasks better (faster, more accurately, etc.) as a result of changes produced by the learning process. Except for this basic consensus on what it means to learn, there are precious few assumptions shared by all four paradigms.

The central purpose of this special volume is to acquaint the reader with each machine learning paradigm, and do so directly from the proverbial horse's mouth—that is, as presented by one or more prominent researchers who practice each respective ML approach. Each author was asked to write a self-contained article with explicit historical and cross-paradigmatic perspectives for a reader well informed in artificial intelligence, but not necessarily an expert in machine learning.[1] Most authors complied and produced comprehen-

[1] The articles solicited were submitted to formal review, resulting in significant filtering and in substantial improvements to most of the accepted manuscripts.

sive articles setting forth explicitly many of the assumptions inherent in their machine learning paradigm, the basic computational methods, and the evolution of these methods up to and including reports on the authors' latest research results. If this special volume also serves the role of improving communication between practitioners of different paradigms of machine learning by encouraging cross-comparisons and providing better comprehension of different means to achieve common aims, so much the better.

2. The Inductive Paradigm

The most widely studied method for symbolic learning is one of inducing a general concept description from a sequence of instances of the concept and (usually) known counterexamples of the concept. The task is to build a concept description from which all the previous positive instances can be rederived by universal instantiation but none of the previous negative instance (the counterexamples) can be rederived by the same process. At this level of abstraction, the problem may sound simple, but it is not even well posed. The design space of potential inductive systems is determined by many important dimensions, such as:

– *Description language.* The language in which input instances and output concepts are expressed can vary in representational power (e.g., propositional calculus, first-order logic, or beyond), in whether the domain of variables in the description language is discrete, continuous or mixed, and in whether individual values are points in the domain or probability distributions among the possible domain values. Most early concept acquisition systems handled only certain classes of propositional representations (attribute-value lists) with single-valued variables drawn from a finite nominal domain. Continuous variables were arbitrarily partitioned into discrete intervals. Present systems explore the full range of possibilities. However, most systems make a fixed vocabulary assumption in that all the relevant descriptors must be present at the outset. Lately, some researchers are starting to consider the implications of description languages that grow during the learning cycle, labeling the process *representational shift.*

– *Noise and instance classification.* Most early learning-from-examples systems assumed that every instance was correctly classified as positive or negative with respect to the desired concept; that is, they assumed a benign and accurate teacher providing a stream of well-formed data [16]. Since such an assumption is much too restrictive for real-world applications, new systems explore the possibility of inaccurately labeled and unlabeled instances, of partially specified instances (where some attributes may be unknown), of measurement errors in the values of the attributes, and of differential relevance among the attributes. So long as the signal-to-noise ratio is acceptable, and the

number of instances is sufficiently high, statistical techniques integrated into the learning method come to the rescue.

– *Concept type.* Some learning systems strive for *discriminant concepts*, where the concept description is a set of tests which separate all instances of the concept apart from all instances of every other concept known to the system. Often discriminant concept descriptions are encoded as paths from the root to the leaves of incrementally acquired decision trees. Other learning systems acquire *characteristic concepts*, which strive for compactness and elegance in the concept descriptions. Such concepts are far easier to communicate to human users and often prove more usable when they must be interpreted by some other part of the performance system. However, the tradeoff for simplicity of description is often loss of complete accuracy; characteristic concepts do not necessarily comply with the strict discrimination criterion. Characteristic concept descriptions are often encoded as frames or logical formulae. The *inductive bias* of a learning system is often expressed as preferences in the type of concept to be acquired, and simplicity of the concept description is the most prevalent form of domain-independent inductive bias.

– *Source of instances.* The initial learning-from-examples model called for an external teacher to supply a stream of classified examples for a single concept to be acquired at one time. In addition to considering the possibility of noise in the data (discussed above), one can remove the teacher entirely and use the external world as a source of data. In such cases, the learner must be proactive in seeking examples, must cope with multiple concepts at one time, and must seek its own classification of instances by appealing to an external oracle (if available), by performing experiments (if possible), or by conceptual clustering techniques [10]. Current work also addresses the judicious selection of instances to reduce maximally the uncertainty in partially formed concepts (a complex form of multi-dimensional binary search).

– *Incremental versus one-shot induction.* One-shot inductive learning systems consider all the positive and negative instances that will ever be seen as training data at one time and produce a concept description not open to further modification [4]. Incremental techniques produce the best-guess concept [16] or the range of concepts consistent with the data so far (as in version spaces [12]), and can interleave learning and performance. As the latter reflect more accurately real-world situations in which learning is an ongoing process, they are currently the ones more heavily investigated.

3. The Analytic Paradigm

A more recent but very widely studied paradigm for learning is based on analytical learning from few exemplars (often a single one) plus a rich underlying domain theory. The methods involved are deductive rather than

inductive, utilizing past problem solving experience (the exemplars) to guide which deductive chains to perform when solving new problems, or to formulate search control rules that enable more efficient application of domain knowledge. Thus, analytic methods focus on improving the efficiency of a system without sacrificing accuracy or generality, rather than extending its library of concept descriptions. The precursors of modern analytic learning methods are macro-operators [5], and formal methods such as weakest precondition analysis. Presently, analytic learning methods focus on explanation-based learning [3, 13], multi-level chunking [9], iterative macro-operators [2] and derivational analogy [1]. Some fundamental issues cut across all analytic methods:

– *Representation of instances.* In analytic methods an instance corresponds to a portion of a problem solving trace, and learning uses that single instance plus background knowledge (often called a *domain theory*). In the simplest case an instance is just a sequence of operators, which can be grouped into macro-operators, modified in analogical transfer, or viewed as steps in a "proof" of the problem solution for explanation-based learning. More recently, problem solving traces carry with them the justification structure (i.e., the goal-subgoal tree, annotations on why each operator was selected, and a trace of failed solution attempts, all interconnected with dependency links). These traces permit richer learning processes such as generalized chunking, derivational analogy (applied by Mostow in this volume) and explanation-based specialization (discussed by Minton et al. in this volume).

– *Learning from success or failure.* The earliest analytic techniques acquired only the ability to replicate success more efficiently (e.g., macro-operators, early EBL, and early chunking). However, much can be learned from failure in order to avoid similar pitfalls in future situations sharing the same underlying failure causes. Recent EBL techniques, analogical methods, and to some extent chunking in systems like SOAR [9] learn both from success and from failure.

– *Degree of generalization.* The control knowledge acquired in analytical learning can be specific to the situation in the exemplar or generalized as permitted by the domain theory. Generalization strategies range from the elimination of irrelevant information (in virtually all analytical methods) to the application of general meta-reasoning strategies to elevate control knowledge to the provably most general form in the presence of a strong domain and architectural theory (as discussed by Minton et al. in this volume).

– *Closed versus open loop learning.* Open loop learning implies one-pass acquisition of new knowledge, regardless of later evidence questioning its correctness or utility. In contrast, closed loop learning permits future evaluation of the new knowledge for modification or even elimination should it not improve system performance as desired. Performance measures of newly

acquired knowledge are often empirical in nature; only the acquisition of the control knowledge is purely analytical.

4. The Genetic Paradigm

Genetic algorithms (also called "classifier systems") represent the extreme empirical position among the machine learning paradigms. They have been inspired by a direct analogy to mutations in biological reproduction (cross-overs, point mutations, etc.) and Darwinian natural selection (survival of the fittest in each ecological niche). Variants of a concept description correspond to individuals of a species, and induced changes and recombinations of these concepts are tested against an objective function (the natural selection criterion) to see which to preserve in the gene pool. In principle, genetic algorithms encode a parallel search through concept space, with each process attempting coarse-grain hill climbing.

Stemming from the work of Holland [7], the genetic algorithm community has grown largely independent of other machine learning approaches, and has developed its own analysis tools, applications, and workshops. However, many of the underlying problems and techniques are shared with the mainline inductive methods and with the connectionist paradigm. For instance, as in all empirical learning, assigning credit (or blame) for changes in performance as measured by the objective function is difficult and indirect. There are a multiplicity of methods to address this problem in the inductive approaches, dating back to Samuel [15]. For genetic algorithms, Holland developed the *bucket brigade* algorithm [8]. And, credit/blame assignment is positively central to all connectionist learning methods, as exemplified by the backpropagation technique.

5. The Connectionist Paradigm

Connectionist learning systems, also called "neural networks" (NNets) or "parallel distributed systems" (PDPs), have received much attention of late. They have overcome the theoretical limitations of perceptrons and early linear networks by the introduction of "hidden layers" to represent intermediate processing and compute nonlinear recognition functions. There are two basic types of connectionist systems: those that use distributed representations—where a concept corresponds to an activation pattern spanning, potentially, the entire network—and those that use localized representations where physical portions of the network correspond to individual concepts. The former is the more prevalent, although hierarchical modularization for complex systems limits the physical extent of concept representations.

Connectionist systems learn to discriminate among equivalence classes of patterns from an input domain in a holistic manner. They are presented with training sets of representative instances of each class, correctly labeled (with

some noise tolerance), and they learn to recognize these and other instances of each representative class. Learning consists of readjusting weights in a fixed-topology network via different learning algorithms such as *Boltzmann* [6] or *backpropagation*. These algorithms, in essence, calculate credit assignment from the final discrimination back to the individual weights on all the active links in the network. There are, of course, much more complexity and many subtle variations involved, as reported in Hinton's article in this volume.

Amidst structural diversity, one can find strong functional similarities between connectionist learning systems and their symbolic counterparts, namely discriminant learning in inductive systems and genetic algorithms. Induced symbolic decision trees and NNets both are trained on a number of pre-classified instance patterns, both are noise-tolerant, and after training both are given the task of classifying new instances correctly. In order to evaluate the appropriateness of each technique to the task at hand, one must ask some detailed quantitative questions, such as comparing the ease of casting training data into acceptable representations, the amount of training data required for sufficiently accurate performance, the relative computational burden of each technique in both training and performance phases, and other such metrics.

6. Cross-Paradigmatic Observations

Consider the larger picture, contrasting the three symbolic paradigms and connectionist systems in general. But, rather than engaging in the perennial sectarian debate of supporting one paradigm at the expense of the other, let us summarize the properties of a domain problem that favor the selection of each basic approach:

– *Signal-symbol mapping*. From continuous signals such as wave forms into meaningful discrete symbols such as phonemes in speech recognition. Best approach: Connectionism (or traditional statistical learning methods such as dynamic programming or hidden Markoff models).

– *Continuous pattern recognition*. From analog signals to a small discrete set of equivalence classes. Best approach: Connectionism. Inductive or genetic approaches require that the signal-symbol map be solved first, or that a predefined feature set with numerical ranges be given a priori.

– *Discrete pattern recognition*. From collections of features to membership in a predefined equivalence class (e.g., noninteractive medical diagnosis). Best approach: Inductive learning of decision trees. Other inductive approaches, genetic algorithms, and even connectionist methods can apply.

– *Acquiring new concept descriptions*. From examples to general descriptions. Best approach: Induction with characteristic concept descriptions, per-

mitting explanation to human users or manipulation by other system modules. Genetic algorithms and connectionist approaches do not produce characteristic concept descriptions.

– *Acquiring rules for expert systems.* From behavioral traces to general rules. If a strong domain theory is present, analogical or EBL approaches are best. If not inductive or genetic approaches prevail. Connectionist systems do not preserve memory of earlier states and therefore cannot emulate well multi-step inferences or deductive chains.

– *Enhancing the efficiency of rule-based systems.* From search guided only by weak methods to domain-dependent focused behavior. Best approach: Analytic techniques ranging from macro-operators and chunking to EBL and analogy. Here is where background knowledge can be used most effectively to reformulate control decisions for efficient behavior by analytic means.

– *Instruction and symbiotic reasoning.* From stand-alone system to collaborative problem solving. When user and system must pool resources and reason jointly, or when either attempts to instruct the other, knowledge must be encoded in an explicit manner comprehensible to both. Best approaches: Inductive (with characteristic concept descriptions) or analytic (often case-based analogical) reasoning. Neither genetic systems nor (especially) connectionist ones represent the knowledge gained in a manner directly communicable to the user or other system modules. Imagine attempting to understand the external significance of a huge matrix of numerical connection strengths.

– *Integrated reasoning architectures.* From general reasoning principles to focused behavior in selected domains. In principle all methods of learning should apply, although the analytic ones have been most successful thus far.

At the risk of oversimplification, one may make a general observation: Connectionist approaches are superior for single-step gestalt recognition in unstructured continuous domains, if very many training examples are present. At the opposite end of the spectrum, analytic methods are best for well-structured knowledge-rich domains that require deep reasoning and multi-step inference, even if few training examples are available. Inductive and genetic techniques are best in the center of the wide gulf between these two extreme points. Clearly there are many tasks that can be approached by more than one method, and evaluating which might be the best approach requires detailed quantitative analysis. Perhaps more significantly, there are complex tasks where multiple forms of learning should co-exist, with connectionist approaches at the sensor interface, inductive ones for formulating empirical rules of behavior, and analytic ones to improve performance when the domain model is well enough understood.

REFERENCES

1. Carbonell, J.G., Derivational analogy: A theory of reconstructive problem solving and expertise acquisition, in: R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* **2** (Morgan Kaufmann, Los Altos, CA, 1986).
2. Cheng, P.W. and Carbonnel, J.G., Inducing iterative rules from experience: The FERMI experiment, in: *Proceedings AAAI-86*, Philadelphia, PA (1986) 490–495.
3. DeJong, G.F. and Mooney, R., Explanation-based learning: An alternative view, *Mach. Learning* **1** (1986) 145–176.
4. Dietterich, T.G. and Michalski, R.S., A comparative review of selected methods for learning structural descriptions, in: R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Tioga, Palo Alto, CA, 1983).
5. Fikes, R.E. and Nilsson, N.J., STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2** (1971) 189–208.
6. Hinton, G.E., Sejnowski, T.J. and Ackley, D.H., Boltzmann machines: Constraint satisfaction networks that learn, Tech. Rept. CMU-CS-84-119, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA (1984).
7. Holland, J., *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, MI, 1975).
8. Holland, J.H., 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 (Eds.), *Machine Learning: An Artificial Intelligence Approach* **2** (Morgan Kaufmann, Los Altos, CA, 1986) 593–624.
9. Laird, J.E., Rosenbloom, P.S. and Newell, A., Chunking in Soar: The anatomy of a general learning mechanism, *Mach. Learning* **1** (1986) 11–46.
10. Michalski, R.S. and Stepp R.E., Learning from observation: Conceptual clustering, in: R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Tioga, Palo Alto, CA, 1983).
11. Minton, S., Carbonell, J.G., Etzioni, O., Knoblock, C.A. and Kuokka, D.R., Acquiring effective search control rules: Explanation-based learning in the PRODIGY system, in: *Proceedings Fourth International Workshop on Machine Learning*, Irvine, CA (1987) 122–133.
12. Mitchell, T.M., Version spaces: An approach to concept learning, Ph.D. Dissertation, Stanford University, Stanford, CA (1978).
13. Mitchell, T., Keller, R. and Kedar-Cabelli, S., Explanation-based generalization: A unifying view, *Mach. Learning* **1** (1986) 47–80.
14. Quinlan, J.R., Learning efficient classification procedures and their application to chess end games, in: R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Tioga, Palo Alto, CA, 1983).
15. Samuel, A.L., Some studies in machine learning using the game of checkers, in: E.A. Feigenbaum and J. Feldman (Eds.), *Computers and Thought* (McGraw-Hill, New York, 1963) 71–105.
16. Winston, P., Learning structural descriptions from examples, in: P. Winston (Ed.), *The Psychology of Computer Vision* (McGraw-Hill, New York, 1975).