Machine Learning, 4, 251-254 (1989)

Can Machine Learning Offer Anything to Expert Systems?

BRUCE G. BUCHANAN
Professor of Computer Science, Medicine, and Philosophy, University of Pittsburgh, Pittsburgh, PA 15260

Today's expert systems have no ability to learn from experience. This commonly heard criticism, unfortunately, is largely true. Except for simple classification systems, expert systems do not employ a learning component to construct parts of their knowledge bases from libraries of previously solved cases. And none that I know of couples learning into closed-loop modification based on experience, although the SOAR architecture [Rosenbloom and Newell 1985] comes the closest to being the sort of integrated system needed for continuous learning. Learning capabilities are needed for intelligent systems that can remain useful in the face of changing environments or changing standards of expertise. Why are the learning methods we know how to implement not being used to build or maintain expert systems in the commercial world?

Part of the answer lies in the syntactic view we have traditionally taken toward learning. Statistical techniques such as linear regression, for example, are "knowledge-poor" procedures that are unable to use knowledge we may bring to the learning task. However, learning is a problem-solving activity. As such, we can and should analyze the requirements and functional specifications, the input and output, and the assumptions and limitations. If there is one thing we have learned in AI it is that complex, i.e., combinatorially explosive and ill-structured, problems often can be tamed by introducing knowledge into an otherwise syntactic procedure.

There is not a single model for learning effectively, as is confirmed by the proliferation of methods, just as there is not a single model of diagnosis or of scheduling. If the machine learning community is going to make a difference in the efficiency of building commercial expert systems, we need not only more methods but a better understanding of the criteria for deciding which method best suits which learning problem. No matter what model one adopts, there is a significant chance of success in coupling a learning program with an expert system. And there will be significant benefits to the machine learning community in terms of research problems that crystallize out of the solution of applications.

The workhorse of knowledge acquisition for commercial expert systems is still knowledge engineering. It is labor intensive and prone to error because it involves discussions among persons with very different backgrounds. We are slightly better at teaching knowledge engineering, and practicing it, than we were a decade ago. It has the outstanding virtue that it works. Partly because it is slow, it also allows more deliberate examination of conceptual frameworks, of assumptions, and of mistakes. But it is too slow: it stands in the same relation to knowledge acquisition as do paper and pencil to calculation.

Speeding up the knowledge engineering process has led to the development of customized editors of the SALT variety [Marcus and McDermott 1989]. These are being used successfully and provide some assistance to knowledge base developers. But they are not learning programs; they are editors. As an aside, though, they allow us to meet McCarthy's 1958 challenge [McCarthy 1958] of building programs that can accept advice before we build systems that learn on their own. In a customized editor we see elements of expert systems. They use knowledge of a generic task such as diagnosis to guide the acquisition and debugging of a knowledge base. With our increased experience in building systems over the last decade, we should now be able to integrate the techniques of TEIRESIAS [Davis 1982] and ROGET [Bennett 1985] for even more powerful editors.

As we move closer to automated knowledge acquisition and away from knowledge engineering, we must understand better how to convey the built-in assumptions of the programs that assist in constructing knowledge bases. With expert systems, there is a potential danger that the designers of programs will have different conceptual frameworks in mind than the users. In the case of learning programs, the same potential danger arises in the mismatch of frameworks between the designers of the learning programs and the designers of the expert systems. We in the machine learning community have not adequately addressed this question, nor would we think of it unless there were a pressing need to apply machine learning techniques to problems of interest to another community.

Inductive learning has received considerable attention since the 1950s, with several approaches now in our growing toolkit of programs that can assist in knowledge acquisition. Some programs assume the data are correct, that the descriptions and classifications of examples are error-free; some of the same ones assume that the examples are all available at once, and hereafter remain available; others assume that classification rules are categorical; most assume that the rules to be learned are one-step conjunctive rules. Successful applications of ID3 and C4 [Quinlan, et al. 1986] and their cousins have provided knowledge bases for working expert systems whose task is to classify. These, together with a few other isolated cases of an expert system being built with the assistance of a learning program, have kept alive the promise of payoff from the 35-year investment in learning research.

Explanation-based learning (EBL) is now receiving even more attention than induction or similarity-based learning. Here, more than with induction, the problem-solving nature of acquiring new knowledge is apparent. (See, for example, [Mitchell 1986].) Its main assumption is that enough theory exists to provide a rationalization of why one instance is or is not a prototypical member of a class. What else? Does it have to be a formal, deductive rationalization, or will an empirical argument suffice? (See [Smith, et al. 1985] for an argument that a weaker, empirical model will suffice.) Unfortunately, EBL has not found any applications in building expert systems, thus it is still demonstrated only in laboratory prototypes.

Case-based and analogical reasoning are still pipe dreams when matched against the harsh standards of robustness of commercial applications. Some of the problems stem from the dilemma that we can find some mappings from any previous case or previously studied domain to the present one: we almost have to know the answer in order to find the case or the analogy that lets us program a machine to find the answer. Analogical reasoning involves at least two problem-solving steps: finding a useful analogous domain (or object or process) and finding relevant mappings between the analog and the original thing. It is not a robust method in the sense that it still produces too many false positive results. Moreover, we don't see many suggestions for making these methods more selective. Their power seems to lie in offering suggestions when we have run out of other ideas.

What about neural nets? They certainly are popular because the idea of getting something for nothing has always held great appeal. They have shown some success in knowledge-poor tasks, such as character recognition and other perceptual tasks. Expert systems, by their very nature, are knowledge-intensive, however, and thus are less amenable to learning with syntactic reinforcement methods. Once a neural net is tuned, there is no way to understand why it succeeds on some cases and fails on others except to say that the weights on some nodes are higher than on others. Moreover, their construction is not free; considerable effort must be invested in laying out the structure of the network before examples can be presented.

Only a few of the techniques in the literature are immediately useful, and these have their limits. Part of our research charge needs to include understanding the scope and limits of different methods and determining when they are applicable. Expert systems provide a strong rationale for continued funding of research on machine learning, but they also serve to sharpen our understanding of problems.

Expert systems offer a focus for development of new machine learning methods and better understanding of old ones. Of course, we need basic research as well as demand-driven research. At the moment, however, there is an imbalance in the amount of work on learning in domains where we do not need to learn and on techniques with crippling assumptions. Let us attempt to understand the context of the commercial, military, and medical needs. In research on machine learning, as on other problem-solving methods, new wrinkles—perhaps new opportunities—will arise in experimenting with real, complex knowledge bases and applications.

Using an expert system as a testbed offers a tough test of success. The commercial world of expert systems at large seems unconvinced that machine learning has anything to offer yet. I strongly disagree. Inductive learning, at least, is already in limited use, and present methods can be extended to make them more useful. Some of the issues that need to be resolved in order to make inductive methods a mainstay of commercial knowledge acquisition are already modestly well understood: learning in the context of noisy and uncertain data, exploiting an existing partial theory, representing the objects of learning in a form other than classification rules, and tailoring the learning to the specific context in which the learned information will be used. These are partly issues of utility, but they are important research problems as well. Machine learning is ready for development now, with attendant benefits to us in crystallizing current and new research problems.

References

Bennett, J.S. 1985. ROGET: A knowledge-based system for acquiring the conceptual structure of a diagnostic expert system. *J. Automated Reasoning 1*, 49-74.

Davis, R. 1982. TEIRESIAS: Applications of meta-level knowledge. In R. Davis and D. Lenat (Eds.). 1982. *Knowledge-based systems in artificial intelligence*. New York: McGraw-Hill.

Marcus, S. and McDermott, J. 1989. SALT: A knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence 39,* 1–37.

McCarthy, J. 1968. Programs with common sense. *Proc. Symposium on the Mechanisation of Thought Processes.* Reprinted in M. Minsky (Ed.), *Semantic Information Processing.* Cambridge, MA: MIT Press, 1968.

Mitchell, T.M., Keller, R., and Kedar-Cabelli, S. 1986. Explanation-based generalization: A unifying view. *Machine Learning 1,* 47–80.

Quinlan, J.R., Compton, P.J., Horn, K.A., and Lazarus, L. 1986. Inductive knowledge acquisiton: A case study. *Proceedings Second Australian Conference on Applications of Expert Systems.* In J.R. Quinlan (Ed.), *Applications of Expert Systems.* Maidenhead: Academic Press (in press).

Rosenbloom, P.S., and Newell, A. 1985. The chunking of goal hierarchies: A generalized model of practice. In R. Michalski, J. Carbonell, and T. Mitchell (Eds.), *Machine learning: An artificial intelligence approach (Vol. 2).* Los Altos, CA: Morgan-Kaufmann.

Smith, R.G., Winston, H., Mitchell, T., and Buchanan, B.G. 1985. Representation and use of explicit justification for knowledge base refinement. In *Proceedings of IJCAI85.* Los Altos, CA: Morgan-Kaufmann.