What has learning got to do with expert systems?

WHAT HAS LEARNING GOT TO DO WITH EXPERT SYSTEMS?

Alan Bundy

D.A.I. RESEARCH PAPER NO. 214

Invited talk at the *First Expert Systems Theme Workshop*,
5-6 March, 1984.

What has Learning got to do with Expert Systems?

by
Alan Bundy

Acknowledgements

I am grateful for feedback and suggestions from Richard Young and Bernard Silver. This research was supported by SERC grant, number GR/C/20826.

Abstract

This paper was prepared as an invited talk on learning for the first workshop on expert systems of the IKBS section of the Alvey Programme. It discusses: the significance of learning for expert systems, the tools available and the medium term goals of learning research. This discussion centres around rule and concept learning. It concludes that the significance of automatic knowledge acquisition may have been overstated, but that there are other significant applications of learning in expert systems. The tradeoffs between classification and focussing based learning systems are discussed and some of the outstanding problems are listed, especially the dependence on human provision of the description space and the potential for combinatorial explosions.

1. Introduction

The purpose of this paper is to give a personal opinion on the following questions.

- What is the significance of AI research on learning for expert systems?
- What tools/techniques can the learning research community make available to the expert systems development community, both immediately and potentially?
- What should be the medium term research goals of the learning community, with special reference to the needs of expert systems?

2. The Significance of Learning to Expert Systems

2.1. Knowledge Acquisition

Learning has come to be associated with expert systems research through the problem of knowledge elicitation. The conventional wisdom runs as follows.

1. The incorporation of human knowledge into an expert system shell (knowledge elicitation) is a bottle neck in the building of expert systems.

2. AI learning techniques offer a mechanism for the automatic acquisition of knowledge from a body of experience.

3. Maybe the process of knowledge elicitation can be automated using these learning techniques (knowledge acquisition).

4. Since the classic expert-systems consist of a set of domain-specific production rules in a domain independent shell, we should look particularly at the techniques for learning rules from examples.

I have never really believed this conventional wisdom, and the expert systems workshop has given me a forum to examine it in detail. I see problems with both steps 1 and steps 4 of the argument.

The claim that knowledge elicitation was a bottle-neck in the production of expert systems came first from the mecca of expert systems, the Stanford Heuristic Programming Project. However, from the same source came simultaneous, development-time estimates for expert system products of a few man-months. For instance, there is a claim in [Feigenbaum 79] that the PUFF system required less than 50 hours of interaction with an expert and less than 10 man-weeks of programming effort. Either, Feigenbaum's notion of what constitutes a bottle-neck in AI is radically different from mine, or there is something he is not telling us.

By a simple expert system I mean one which is built by taking a standard expert systems shell, e.g. EMycin, SAGE, etc. and adding some rules. The strength of claim 4 depends on whether simple expert systems can also be products, i.e. make money. The proceedings of the recent SGES conference on expert systems, [Fox 83], seemed to come to two views on this.

(a) On the one hand several newcomers to the field reported success in quickly building simple expert system products. Note that these are all products for in-house use by an organisation, and were usually claimed to be saving rather than making money, i.e. no-one was selling a product' on the open market, apart from the shells themselves. (See e.g. papers by Keen; Maclean & Wilde; and Jackson.)

(b) On the other hand, more experienced groups reported dissatisfaction with existing shells. They had found the need to change the shells: either by altering the inference mechanism, or by allowing rules to trigger arbitrary computation, or in other ways. I will call such systems, **complex expert systems**. These groups had usually ended up building their own shells in order to be able to manipulate them, and were now advocating not shells but toolkits, to facilitate customization. (See e.g. papers by Kidd & Cooper; Rawlings & Fox; Alvey; Merry; Fox, Alvey & Myers; and Jackson.)

Presumably, neither group was lying, so both positions are right. I can only reconcile them by assuming that first approximation solutions to a number of problems are offered by simple expert systems, but that second approximation solutions require complex expert systems. Shells seem to offer a good starting point for newcomers to the field, but one they rapidly become dissatisfied with.

Clearly, the current state of the learning art can only support automatic knowledge acquisition for simple expert systems. So current knowledge acquisition tools are only likely to satisfy people in the short period between becoming interested in expert systems and learning to build their own systems from shells as a prelude to learning how to make their own shell.

However, there is one type of problem for which knowledge acquisition tools may be of more practical use, namely when the human expertise in an area is not well developed. Knowledge acquisition systems may then come up with a better set of rules than could be extracted from a human expert. The classic work in this vein is Michalski's on soya bean diseases, [Michalski et al 82]. A disadvantage of this technique for rule extraction is that the resulting rules may be too complex to be readily understood by human users. Note also that the technique is still dependent on humans to supply the description space for rule formation.

2.2. Other Uses of Rule Learning

The knowledge acquisition use of rule learning techniques is mainly concerned with the learning of factual information. However, many rule learning techniques are equally appropriate for learning both factual and control information, [Bundy et al 84]. By changing the hypotheses of existing rules their control behaviour can be improved. With the aid of Mitchell's solution extraction technique, [Mitchell et al 81], this process could be largely automated, so that an existing rule base could be made more efficient in use.

Many authors prefer the separation of control and factual information, rather than their intermixing within rules. The control information can also be represented by a set of rules and rule learning techniques applied to them. Silver's technique of precondition analysis can be used to discover new control rules from single worked examples, [Silver 83].

The direct use of learning techniques in expert systems has received surprisingly little attention. By direct I mean the use of automatic learning techniques as a central part of an expert system rather than just to provide rules for use by an expert system. For example, rule learning techniques might be used to induce a rule representing the connection between particular foods and an allergy. Examples would consist of food eaten and the state of the patient, e.g. occurrence of migraine, amount of eczema, etc. Note that the rule is an end in itself, rather than an input to anything. The ambiguity of possible causes usually means that tedious, exhaustive, specific testing must currently be done to track down the culprit(s). However, a program might be able to use the normal diet of the patient as sufficient data, using brute force to cope with the ambiguity. Similar applications readily suggest themselves.

3. Learning Tools for Expert Systems

A comparative analysis of some of the AI, rule-learning techniques is given in [Bundy et al 84]. This picks out two main families of techniques: focussing, [Young et al 77], and classification, [Quinlan 79]. Both are essentially techniques for learning concepts from examples and non-examples. Concept learning is applied in rule learning by treating the hypothesis of a rule as the concept to be learnt; examples are situations when the rule correctly fired, and non-examples are situations in which rule either fired when it should not have or did not fire when it should have. Thus the examples and non-examples have to be attached to a particular rule. Rules with disjunctions in their hypotheses can be transformed into several purely conjunctive rules with the same conclusion. So we can imagine the examples and non-examples being attached to a particular conclusion.

The focussing family includes: versions spaces, discrimination and generalization. These techniques are intended for learning conjunctive concepts, and so are limited to rule sets in which there is only one conjunctive rule for each conclusion. This is a serious limitation. It can be lifted at a price.

The problem about multiple rules with the same conclusion is that it is not clear which examples provide evidence for which rules. (Non-examples provide evidence for all rules.) It is possible for the subdivision to be done incorrectly. There are two solutions, either all examples and non-examples must be provided before the learning process begins, or the process must store all the examples and back-up and restart if it encounters a contradiction. The first solution has the drawback that no incremental learning results are available until all the examples have been processed. The second solution has the drawback of a potential combinatorial explosion. Both solutions may require massive storage if there are lots of (non-)examples to be processed.

Classification is the technique use in ID3, [Quinlan 79]. It is inherently for disjunctive concepts and hence multiple rules. It can be adapted to either of the two solutions outlined above, and thus suffers from the drawbacks listed there. Classification is not tuned for conjunctive concepts and can introduce redundant, disjunctive information into them. Techniques exist for removing this.

If only single rules or conjunctive concepts can occur then a cause of choice and search is eliminated from the learning process. If no choices occur then back-up is never necessary. Focussing exploits this possibility by using an incremental learning process and throwing away (non-)examples after they have been processed. However, adapting focussing to multiple rules re-introduces the drawbacks listed above, and so does the introduction of choice points for any of the other reasons listed in the next section.

All rule learning techniques are crucially dependent on the user specifying the description space, i.e. the attributes from which the hypothesis of each rule is formed. Redundant attributes can be supplied; this will merely slow down the learning process as it learns to ignore them. However, the omission of attributes will cause the process to fail.

A commercial product based on classification exists, namely Expert-Ease. Users must supply the attributes for each rule hypothesis, and the values of these attributes for sufficient examples and non-examples for the rule to be learnt. This has been used to build some rule sets for a few simple expert systems.

4. Medium Term Research Goals

Existing rule learning techniques suffer from a number of limitations that need to be addressed. Some have been mentioned above.

Choice points can arise in the learning process for several reasons. The need to back up and remake them is usually indicated because the incremental results of the learning process are contradictory. The contradiction may be caused by any of the following factors.

1. The data may be noisy and the wrong data may have been assumed correct.

2. The blame for a non-example may have been ambiguous and assigned to the wrong attribute.

3. The examples attached to a particular conclusion may have been assigned to the wrong rule.

4. A vital attribute may have been omitted from the description space or the description space may be badly structured.

If any of these factors may be present then it is necessary to save choice points and reuse examples and non-examples. Hence, focussing loses one of its main advantages over classification.

Further research is required to locate the reasons for a contradiction and to control the combinatorial explosion which the choices may cause. There are a number of approaches currently being explored.

- The problem of choosing what (non-)examples to ignore as noise has been tackled from the viewpoint of preferring simple rules to complex ones. Much work has already been done in this area in statistics, [Payne and Preece 80], and we need to learn from this.

- An exploratory learning system might devise experiments for settling ambiguity about blame assignment.

- Analytic techniques have been used to explore the reason for the success/failure of a rule and thus extend the description space appropriately, [Mitchell et al 83].

- The description space might be restructured to simplify the form of disjunctive concepts, e.g. make them conjunctive by the introduction of a new (disjunctive) term.

Current blame assignment techniques (e.g. contradiction backtracing, [Shapiro 81], and solution extraction, [Mitchell et al 81]) are capable only of identifying a faulty instance of a rule, not of locating that fault in the rule (see [Bundy et al 84]). The techniques for identifying factual faults in rules all depend heavily on user intervention or the provision of a standard Tarskian model for the rule set. Location of the fault in the hypothesis of the rule can be done by the modification techniques like focussing or classification. This involves search when the fault could be due to more than one factor (called far-misses in [Bundy et al 84]). There are no techniques for identifying faults in the conclusions of rules.

New techniques are required:

- for removing some of the search from the location of faults;
- for identifying faults in the conclusion of the rules;
- for identifying factual faults without the aid of a model or the intervention of the user.

Techniques for creating new rules are limited by the need for the user to specify the conclusion of the rule and the description space from which its hypothesis must be built. The need to specify the conclusion follows from the lack of techniques of blame assignment for conclusions mentioned above. The need to specify the description space follows from the lack of techniques for automatically extending the description space.

5. Other Areas of Learning

All the discussion so far has been about rule and concept learning techniques. Are there any other areas of learning of relevance for expert systems? The answer is almost certainly yes, but it is difficult to be more precise. The main difficulty is in defining and systematizing the area of learning. Learning is a term we inherit from folk psychology. I doubt it has much application as a natural class of techniques within AI. Consider the following failed attempts to give learning a general AI definition.

- Based on the techniques from rule and concept learning one might be tempted to define learning as the class of techniques for hypothesis formation from data. Unfortunately, this definition would then include line finding techniques from vision and anaphoric reference techniques from natural language.

- Based on techniques from skill acquisition, like Mostow's operationalization, [Mostow 81], one might be tempted to define learning as the class of techniques for transforming a representation from a declarative to a procedural form. Unfortunately, this definition would then include automatic programming transformation and compilers.

- Based on Samuel's hill climbing technique, [Samuel 63], one might be tempted to define learning as the class of techniques for incremental improvement of a procedure or representation. Unfortunately, this definition would then include compiler optimization.

So an attempt to survey other areas of learning runs into the danger of spreading to the whole of AI and computer science. A task I shrink from in this short paper.

6. Conclusions

Limiting our attention to rule and concept learning techniques we see that they do have significance for expert systems. The significance of automatic knowledge acquisition has, perhaps, been overstated, but there are other important applications: in overcoming a lack of human expertise, in improving the search behaviour of large rule sets, and in direct use of learning techniques.

Existing techniques, like focussing and classification, are still limited. The main outstanding problems are the need for human provision of the description space and the lack of techniques for making the choices caused by noisy data and blame assignment ambiguity.

References

[Bundy et al 84] Bundy, A., Silver, B. and Plummer, D.
An Analytical Comparison of some Rule Learning Programs.
Research Paper, Dept. of Artificial Intelligence, Edinburgh, 1984.
Submitted to Artificial Intelligence Journal. Earlier Versions in Proceedings of ECAI-82 and in Proceedings of the Third Annual Technical Conference of the British Computer Society's Expert Systems Specialist Group.

[Feigenbaum 79] Feigenbaum, E. A.
Themes and case studies of Knowledge Engineering.
In Michie, D. (editor), *Expert systems in the micro-electronic age*, pages 3-25. Edinburgh University Press, 1979.

[Fox 83] Fox, J. (editor).
Expert Systems 83.
BCS Special Interest Group on Expert Systems, 1983.

[Michalski et al 82]
Michalski, R.S., Davis, J.H., Bisht, V.S. and Sinclair, J.B.
Plant/ds: An expert consulting system for the diagnosis of soybean diseases.
In *Proceedings of ECAI-82*, pages 139-140. ECAI, 1982.

[Mitchell et al 81]
Mitchell, T.M., Utgoff, P. E., Nudel, B. and Banerji, R.
Learning problem-solving heuristics through practice.
In *Proceedings of IJCAI-81*, pages 127-134. International Joint Conference on Artificial Intelligence, 1981.

[Mitchell et al 83]
Mitchell, T.M., Utgoff, P. E. and Banerji, R.
Learning by Experimentation: Acquiring and modifying problem-solving heuristics.
In Michalski, R.S, Carbonell, J.F. and Mitchell, T.M. (editors), *Machine Learning*, pages 163-190. Tioga Press, 1983.

[Mostow 81] Mostow, D.J.
Mechanical transformation of task heuristics into operational procedures.
PhD thesis, Carnegie-Mellon University, 1981.

[Payne and Preece 80]
Payne, R.W. and Preece, D.A.
Identification Keys and Diagnostic Tables: a Review.
Journal of the Royal Statisical Society, Series A 143(3):253-292, 1980.

[Quinlan 79] Quinlan, J.R.
Discovering Rules by Induction from Large Collections of Examples.
In Michie D (editor), *Expert Systems in the Micro-Electronic Age,* pages 168-201. Edinburgh University Press, 1979.

[Samuel 63] Samuel, A.L.
Some studies in machine learning using the game of checkers.
In Feigenbaum, E. and Feldman, J. (editors), *Computers and Thought,* pages 71-105. McGraw Hill, 1963.

[Shapiro 81] Shapiro, E.
An algorithm that infers theories from facts.
In *Proceedings of IJCAI-81,* pages 446-451. International Joint Conference on Artificial Intelligence, 1981.

[Silver 83] Silver, B.
Learning Equation Solving Methods from Examples.
In Bundy, A. (editor), *Proceedings of the Eighth IJCAI,* pages 429-431. International Joint Conference on Artificial Intelligence, 1983.
Also available from Edinburgh as Research Paper 184.

[Young et al 77] Young, R.M., Plotkin, G.D. and Linz, R.F.
Analysis of an extended concept-learning task.
In Reddy, R. (editor), *Proceedings of IJCAI-77,* pages 285. International Joint Conference on Artificial Intelligence, 1977.