The Knowledge Engineering Review

Capturing expertise by rule induction

BRIAN P. BLOOMFIELD
Systems Group, The Open University, Milton Keynes, MK7 6AA

Abstract

This paper examines the claim that machine induction can alleviate the current knowledge engineering bottleneck in expert system construction. It presents a case study of the rule induction software tool known as Expert-Ease and proposes a set of criteria which might guide the selection of appropriate domains.

1 Introduction

It is widely accepted that one of the critical bottlenecks in the construction of expert systems lies in the elicitation and representation of expert knowledge, which can typically take months of intensive effort. The reasons for this stem largely from the fact that when knowledge elicitation takes the form of in-depth interviews, human experts often find it difficult to articulate and make explicit the knowledge that they undoubtedly have [22] indeed, much expertise is held on a tacit basis [15]. Thus, even though experts are able to articulate certain formal rules, heuristics, and known facts of their domain, the information they leave out can be more important because of its crucial role within their overall expertise; this unstatable knowledge sets the context within which overt knowledge is exercised [2]. In order to circumvent this problem there have been moves in certain quarters of AI to use an alternative knowledge elicitation technique known as machine induction.[1] This involves the automatic induction of domain rules from examples of decisions (e.g. classifications or diagnoses) provided by experts. The rationale behind this approach is the contention that while human experts may be poor at articulating domain rules suitable for direct inclusion in the rule base of an expert system, they can readily supply examples of their decisions. Building on the work of Hunt [8], several machine-efficient programs now exist to aid the speedy construction of expert system rule bases [1, 10, 12, 16]. One of these programs—Quinlan's ID3 algorithm—is now commercially available in several packages such as Expert-Ease which is produced by Intelligent Terminals Ltd see: [14, 18, 21]. Given a series of domain examples (known as a training set) for which values of specific attributes pertaining to the examples are assigned, the program can induce an exhaustive decision tree or machine-executable classification rule for those examples.

The use of programs to induce machine-executable rules from domain examples raises a number of important philosophical issues; for instance, we might question the validity of inferences based upon only incomplete or partial knowledge of a domain. The problem of induction—whether by humans or machines—has occupied philosophers for a considerable time and remains essentially unresolved. Michalski [11] points out that philosophers such as Polya doubted the possibility of machines ever making plausible inferences. However, Michalski himself takes the view that recent developments in artificial intelligence (AI) enable sufficient background knowledge to be given to a computer program so that the problem of induction by computers can now be considered to have become more tractable. Michalski and Chilausky [12] also claim that domain rules derived by machine induction methods can be more successful in practical trials than those obtained by the conventional interview-based methods of knowledge engineering. These are obviously important claims and indicate that machine induction merits some scrutiny. In this paper I aim to set the technique of machine induction in perspective, and in particular to explore the claim that it can alleviate the bottleneck in knowledge engineering. I will also develop some useful criteria for selecting domains suitable for the elicitation of knowledge by this method.

In the following section I will briefly introduce the idea of induction before going on to discuss the expert system development tool known as Expert-Ease which I will use for a case study of machine induction. (For other critiques of Expert-Ease see: [5, 6, 7, 13]).

2 The problem of induction

Put crudely, the process of induction (inductive inference) involves an attempt to make a general statement about a class of objects based upon particular pieces of information about those objects. For example, on the basis of observing the attribute "whiteness" among several different members of the swan species one might infer that all swans were white. The intrinsic problem with such reasoning is that induction is really a form of conjecture and there cannot exist any formal or logical proof that an inductive inference is actually true; unless, that is, all of the objects in a domain are known. This contrasts with the situation of deductive inference where, providing certain general statements are true, then other more particular statements deduced from them are also true. The quandary posed by the contrast between deductive and inductive inference is neatly summed up by Shapiro [20, chapter 1, p. 5]:

> The reason that it is not in general possible to provide an inverse to the deduction function is that unless all relevant facts are available, a collection of facts is not a replacement for a theory and a theory induced from an incomplete set of facts may always be refuted when further facts come to light.

In principle it is possible to induce an unlimited number of hypotheses about any given collection of objects—a multiplicity which poses severe problems in assessing the validity of such hypotheses. However, the purpose of machine induction is not, strictly speaking, necessarily that of discovering new or original rules about a domain but, rather, that of capturing an adequate representation of the knowledge of an expert. A number of practitioners of machine induction therefore impose the working constraint that induced rules should be both intelligible and plausible to a domain expert. Some also suggest that only the simplest rules should be selected—following the principle of Occam's Razor. By such means it is contended that the problem of the validity of rules is not as severe as might otherwise be supposed. Thus, while the computer can be used as a super-efficient means of generating possible rules, human experts can serve an evaluative function in accepting or rejecting rules when appropriate.

3 Expert-ease

As pointed out earlier, Expert-Ease is an implementation of Quinlan's ID3 algorithm. In its present form Expert-Ease can handle a training set of the order of 250–300 examples; any training set can have up to 31 atrributes with either logical (meaning nominal) or integer values. An example of an integer attribute could be temperature, with values such as 10c, 12c, or 45c; a logical attribute could be represented by the weather, with values such as sunny, raining, or snow. Integer attributes can have any value in the range from $-32,766$ to $+32,767$; while logical attributes can be assigned up to 32 different values. A decision class or category is assigned for each example and can take on a series of logical values. Essentially, the program works by finding an efficient classification rule or decision tree for the examples. It is important to appreciate that Expert-Ease does not induce the only decision tree for a given training set—indeed, different rule induction algorithms will tend to give different results from the same examples. Further, in certain cases one only has to increase the size of a training set by one example in order to force Expert-Ease to arrive at a rule that is substantially different in appearance—e.g. the second rule might involve tests on attributes that were previously untested.

In order to explore some of the issues inherent in attempts to elicit knowledge using Expert-Ease I propose to consider the following questions:

3.a How can suitable domains for the use of rule induction be identified?

While it is possible to show that machine induction is useful in certain well chosen domains such as fault diagnosis or chess end games [17]—where all of the objects in a domain are known—a question mark inevitably hangs over its role in those areas of expertise where knowledge is less well-bounded. The problem that faces knowledge engineers who have to decide whether or not to use a machine induction technique is the fact that the very nature of induction allows for no general theory of applicability; without that knowledge engineers can only hope to arrive at piecemeal criteria by which domains suitable for machine induction might be identified. One such criterion comes from the distributors of Expert-Ease who warn that the program is not suitable for problem domains in which random or stochastic influences play an important role—other criteria will emerge in the context of the following questions and will be properly formulated in the final section.

3.b If experts are often unable to fully explicate their knowledge or to easily provide operating rules of their domain, can they provide an appropriate range of examples (a training set) of their decisions?

Exponents of machine induction argue that although experts may find it difficult to provide domain rules, they can easily supply domain examples. However, though it seems fairly evident that experts can provide examples of their expertise or judgements, this is not to say that they can stipulate an exhaustive set of examples which encapsulate their expertise. That is, the provision of a training set involves much more than giving a list of possibly random examples; rather, these have to be carefully chosen if any really useful knowledge is to be transferred from the expert to the program. And if experts cannot fully explicate the basis of their expertise how can we assume that they are in any better position to provide an adequate training set?

3.c As each example is specified by a decision class or category, and a series of attributes, can experts provide the relevant attributes?

If there is any merit in the argument that experts rely on intuitive know-how rather than rules[4], then what standing do the posited attributes have? This is an important question because the attributes supplied to a rule induction program are perhaps the most crucial input in this form of knowledge elicitation process. Yet, Expert-Ease itself cannot assist in the correct selection of attributes. To be sure, experts can point out salient features pertaining to their decision-making activities—otherwise they would hardly qualify as experts—but how can it be known whether or not these constitute a sufficient set of attributes for the construction of a valid decision tree? A similar point has been noted by Fox [5]; he imagines a case in which a diagnostic decision rule is induced from a training set composed of patients' records that include the attribute "age"—"if all the patients have different ages then ... this is a perfect discriminant for the examples but if, in fact, age is irrelevant to a diagnosis it will also have no predictive power". While a domain expert could spot such obviously spurious attributes there is no guarantee that other more plausible but equally spurious attributes will not be included—experts cannot account for all that they know. To be sure, if a complete set of examples is presented to the program then the resultant decision tree is exhaustive; but this restricts the use of rule induction to domains where all the objects are known beforehand. It is obviously more desirable if induced decision trees have a predictive power for new examples: but this will depend on the epistemological standing of the particular inductions in question.

3.d Given (3.c), can experts unambiguously assign values for the attributes?

Even if, for the sake of argument, we assume that experts are able to supply examples and relevant attributes for a particular domain, can it be further assumed that the possible values that attributes can take on are as readily available too? With logical attributes the values have to be strictly defined in advance. However, an expert might well know that a particular factor was important, and therefore constituted a putative attribute, but be uncertain as to exactly how the values of the factor could be defined. Also, there is a related question here concerning the sensitivity of a decision tree to discrepancies in the assigned values of attributes for any given example. General answers to these questions cannot be given for they will depend on the particular domain and training set in question.

3.e What happens when users encounter new examples? can the values of attributes for any given example always be unambiguously assigned?

The user of a decision tree might employ several different strategies to deal with new examples that potentially have attributes, or specific values for attributes, which do not conform to those specified with the original training set. The existence of such anomalous examples is important because of the problematical nature of inductive inference: inductions based on an incomplete universe of objects are always open to refutation by new instances. It turns out that the reactions of people to objects which are similar to, but also different from what they expect, has been studied by anthropologists and psychologists; these studies have an important bearing on the practical utility of machine induction. For example, one commentator sums up the matter as follows:

> In a chaos of shifting impressions, each of us constructs a stable world in which objects have recognizable shapes, are located in depth, and have permanence. In perceiving we are building, taking some cues, rejecting others. The most acceptable cues are those which fit most easily into the pattern that is being built up. Ambiguous ones tend to be treated as if they harmonised with the rest of the pattern. Discordant ones tend to be rejected [3, p. 36].

The implication of this insight for the use of classification rules is that the treatment of anomalous examples cannot be controlled by the knowledge engineer; different users of a classification rule may react differently to the same anomaly. Where rule induction is used as an aid by domain experts this is not really a problem for they are in the best position to judge what should and should not be considered to be an anomalous example; but it does imply that there is a limit to the level of user expertise below which a classification rule should only be used with caution. In terms of costs, depending on the nature of the problem in hand, the false classification of new examples could have consequences which were more or less disastrous. For many domains the calculus of costs might be such that false classifications (and their subsequent actions) constituted an insignificant, or at least an acceptable risk. However, recently certain exponents of rule induction have called for the use of the technique in relation to tasks such as aircraft take-off and landing and the control of nuclear power installations. In domains such as these the calculus of costs would obviously be an important criterion for assessing the practical utility of induced rules.

3.f What is the nature of a decision tree constructed by Expert-Ease? Put another way, what does it mean to say that the program classifies the training set?

To illustrate the nature of the classification rule or decision tree produced by Expert-Ease, it is useful to consider the following case. Suppose our training set consisted of co-ordinates for the X-Y plane, that each example had an X-value and a Y-value, and that the decision class was either "yes" (indicating that the example was on the straight line Y = X) or "no" (indicating that the example was not on the line Y = X). Table 1 indicates such a possible training set.
Table 2 shows the decision tree induced by Expert-Ease from this training set.[2]

Table 1 Training set of X–Y co-ordinates

X	Y	CLASS	X	Y	CLASS
1	0	no	5	1	no
0	1	no	6	6	yes
1	1	yes	6	7	no
2	1	no	8	9	no
3	3	yes	10	10	yes

Table 2 Decision tree for the training set

```
X
    <9: Y
        <7: Y
            <2: X
                <2: X
                    <1: no
                    ≥1: Y
                        <1: no
                        ≥1: yes
                ≥2: no
            ≥2: yes
        ≥7: no
    ≥9: yes
```

Clearly, in this case Expert-Ease does not produce the most obvious classification rule: if X = Y then yes, if not (X = Y) then no. Rather, in classifying the examples and thereby inducing a decision tree, Expert-Ease effectively partitions the X-Y plane (the attributes space) into areas of "yes" and areas of "no". And if we asked the program to classify other examples not originally included in the training set (thus testing the predictive power of the decision tree) it would classify each example according to which partition its co-ordinates happened to fall into. The accuracy of the decision would depend on the relative size of the training set in comparison to the universe of co-ordinates; the greater the training set, the more fine detail we would have in the decision tree—the smaller the partitions in the X-Y plane—and the more powerful would be its predictive power. In this example the problem of the resolution of the decision tree arises because the attributes had integer values but with logical attributes, the decision trees are somewhat different. As stated earlier, with logical attributes all possible values have to be specified in advance; otherwise, the program would query any new example that had undefined values for an attribute. Such examples could of course be included, but the decision tree would have to be re-induced—which brings us back to the problem of dealing with potential anomalies.

3.g What is the epistemological status of a machine-induced rule?

The most fundamental question of all concerns the relationship between an expert's intuitive knowledge and a rule induced from examples of his or her decisions. More specifically, does an induced rule represent a formal statement or transcription of an intuitive rule of thumb, or does it merely represent a means of accounting for the competence of the expert? Although a machine induced rule might appear reasonable to a domain expert, this does not mean that he or she actually uses that rule in practice—nor indeed, that it is necessarily valid because it is plausible. When experts enter into a dialogue with a knowledge engineer they begin to engage in form of a role-play in which they reflect on their expertise in an abstract and artificial way. When asked to supply domain examples and relevant attributes they implicitly take on board the assumptions of the machine induction method. Further, there is a danger in this that they might tend to say what the knowledge engineer wishes to hear. And, as Waterman [22] notes, experts can be nudged or constrained into a way of thinking that bears little or no relation to their real problem-solving activity. Critics of knowledge engineering such as the Dreyfuses argue that experts do not use rules but, rather, intuitive processes built up through experience; these do not depend on statable attributes so much as whole visual and emotional experiences which are stored in an expert's memory. On this view the stipulation that machine induced rules are required to be intelligible or plausible may not be quite as powerful a control as some have claimed. For if experts do not use rules, or if they don't rely on statable attributes, then any plausibility granted to decision rules may be primitive at best, and completely spurious at worst.

Another means of assessing the epistemological status of machine induced classification rules is to compare them with the standing of other forms of knowledge which are also conjectural in character. One useful comparison can be made with mathematical knowledge; here I will specifically consider the problem of classifying mathematical objects. In seeking to make generalizations about the properties of objects such as polyhedra, mathematicians make inferences in the form of conjectures and then attempt to prove them—if successful, conjectures become accepted as theorems. For example, take the Euler-Descartes conjecture which provides a formula for the classification of polyhedra; it states that for all regular convex polyhedra, $V - E + F = 2$; where V, E, and F represent the number of vertices, edges, and faces respectively [9]. To illustrate the formula, we might consider the case of the simple cube; it has 8 vertices, 12 edges, and 6 faces. Substituting these values in the formula yields the following: $8 - 12 + 6 = 2$. A number of proofs exist for this conjecture, but just as an inductive inference based on an incomplete universe of examples is always open to refutation by new examples, so too the mathematician cannot know whether there exist unknown counterexamples which could refute the conjecture or a proof for it.

Now, what if we specified a training set for Expert-Ease which consisted of examples of potential polyhedra, with attributes V, E, and F, and decision categories to indicate the status of each example—i.e. whether or not the formula held. Clearly, following the implications of the experiment with co-ordinates from the X-Y plane, Expert-Ease could not discover a rule in the form of the Euler-Descartes conjecture. But the epistemological gulf between whatever classification rule Expert-Ease induced and its mathematical counterpart does not simply stem from differences in power and generality: for the Euler-Descartes conjecture is accompanied by proofs—which though fallible, provide a more substantial basis for commitment and usage.

4 Overview

At the beginning I stated the aim of assessing whether or not rule induction might solve the knowledge engineering bottleneck. The preceding section has set out a number of problems in relation to the use of rule induction and depending on how one interprets them one can arrive at different conclusions. In general terms, these problems can be viewed in two different ways. The first way would be to consider them as technical or methodological in nature—in other words, they are concerned essentially with the need to ensure an efficient and meticulous approach to rule induction. The other way would be to view the problems as arising from the differences between the real knowledge of domain experts and the model of expertise underpinning rule induction methods of knowledge elicitation. Of course the latter position does not contradict the conclusion that methodological rigour must become a hallmark of knowledge elicitation by rule induction; indeed it provides a more powerful reason for extolling it. However, it says much more than that; for it implies that assessing the utility of rule induction in any particular domain is a task that depends more on a substantial knowledge of that domain than it does on the adherence to methodological strictures. On the former view, removing the knowledge engineering bottleneck is only a matter of time and effort; on the latter view, either the bottleneck is bound to remain or it will be merely superseded by problems arising from the use of spuriously induced rules in ill-chosen domains.

That the machine induction lobby does not appear to have seriously taken up the issue as to the possible range of reasons why experts are inarticulate stems from their assumptions about the nature of expert knowledge. In a sense it seems fair to say that exponents of machine induction merely wish to devise techniques that will get round the bottleneck problem. That is, the fact that experts cannot fully account for their own expertise is not the point of departure for philosophical or other in-depth analysis; rather, it serves only to stimulate the development and refinement of knowledge elicitation techniques **within** the existing framework or paradigm of knowledge engineering. The problem is not thought to lie in the models of expertise assumed by knowledge engineers but in a lack of sufficiently powerful methods. Among some exponents of machine induction (but also others in AI more generally) this has led to a rather superficial treatment of the problem. For example, Shapiro states that "expertise does not include the ability to explain the reasons for ... professional decisions ... [experts] have to a large extent forgotten how they learned their trade, which tends to be largely based on experience assimilated into a form of intuitive 'Know-how' ". Further, Waterman [22] suggests that expert knowledge is stored in the human mind in a "compiled" form. [For an alternative view to that proposed within AI see: 4, 19].

In a similar vein, advocates of rule induction methods have not adequately considered the epistemological status of induced rules. In short, there may be little basis for faith in an induced rule derived from incomplete data—however plausible it might appear; there is no equivalent to the mathematicians's proof or anything approaching it. Yet, some advocates of machine induction would all but exclude experts from the rule induction process. For example, in cases where data sets already exist (e.g. medical records) there has been at least one suggestion that domain experts—whose time is costly—be bypassed until the later stages of the knowledge engineering process and rules induced directly from raw data. In contrast, in light of the problems outlined earlier, it would seem reasonable to take an opposite view and suggest that domain experts must be brought into the rule induction process early on.

Finally, it is worth returning to the problem of identifying suitable domains for the use of rule induction; on the basis of the preceding discussion the following criteria might be considered to be useful, though by no means exhaustive.

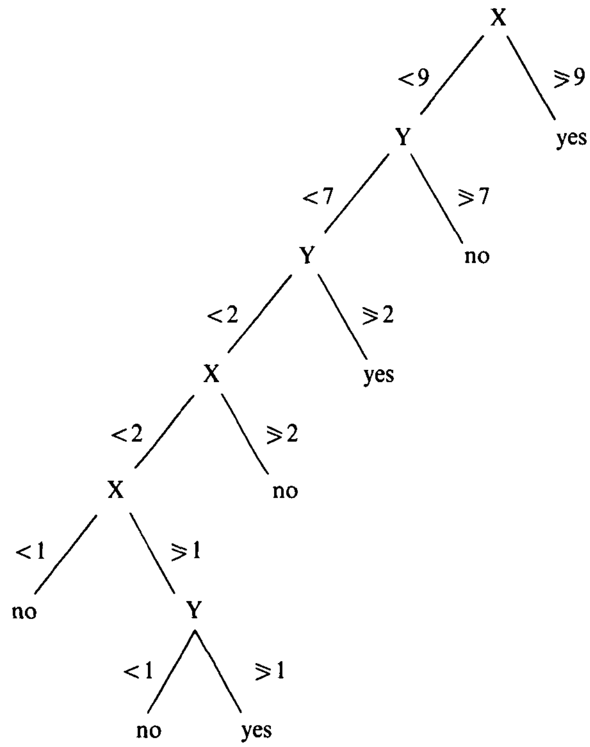
1 Any chosen domain must contain sufficient examples such that it is possible to construct a training set which constitutes a comprehensive encapsulation of expertise in that domain. In other words, domain experts must be able to have confidence in providing the **important** examples.
2 It must be possible to choose attributes that can be clearly defined, both conceptually and in terms of the values that they can take on.
3 Any potential domain should allow the choice of a training set and attribute values which are sufficient to give the resulting decision tree enough resolution to cope with the range of examples likely to be encountered in the future.
4 Domain experts must be able to formulate specific guidelines about how potentially anomalous examples are to be dealt with.
5 Before a decision is taken as to whether or not to proceed with machine induction in a given domain, it is necessary to consider the potential costs of misclassification. These costs can then be used as a criterion for informing the final decision.
6 The "deep" knowledge of the domain should be such that the "surface" plausibility of classification rules can be backed-up by more substantial arguments.

Summary

In this paper I have sought to examine the claim that machine induction can alleviate the bottleneck in knowledge engineering. Focusing on several problems inherent in the use of one particular machine induction tool (Expert-Ease) I have suggested that solving the bottleneck problem is not merely a matter of technique but necessitates a deeper investigation of the nature of human expertise. On a more positive note, I have none the less concluded by suggesting a set of cautious criteria which can be used to guide the selection of domains suitable for the use of Expert-Ease.

Notes

1 Induction is also very important in the area of machine learning.
2 The decision tree in Table 2 can be presented in a more readable form as follows:



References

[1] Brooks, A. and Alty, J.L. (1985) The Use of rule Induction, A Knowledge Acquisition Technique for Expert Systems, to Interpret HCI Experiments, in D. Johnson and S. Cook (Eds.) *People and Computers: Designing the Interface* (Cambridge: Cambridge University Press).

[2] Collins, H.M.; Green, R.H. and Draper, R.C. (1985) Where's the Expertise: Expert Systems as a Medium of Knowledge Transfer, in M.J. Merry (Ed.) *Expert Systems 85* (Cambridge: Cambridge University Press), 325–34.

[3] Douglas, M. (1966) *Purity and Danger* (London: Routledge & Kegan Paul)

[4] Dreyfus, H.L. and Dreyfus, S.E. (1984) From Socrates to Expert Systems, *Technology in Society*, vol. 6, 217–33.

[5] Fox, J. (1984) Doubts About Induction, *Bulletin of SPL Insight*, vol. 2(2), 32–6.

[6] Hart, A. (1984) Experience in the Use of an Inductive System in Knowledge Engineering, in M.A. Bramer (Ed.) *Research and Development of Expert Systems, Proc. of Expert Systems'84* (Cambridge: Cambridge University Press), 117–26.

[7] Hart, A. (1985) The Role of Induction in Knowledge Elicitation, *Expert Systems*, vol. 2(1), 24–8.

[8] Hunt, E.B.; Marin, J. and Stone, P. (1966) *Experiments in Induction* (New York: Academic Press)

[9] Lakatos, I. (1976) *Proofs and Refutations* (Cambridge: Cambridge University Press).

[10] Michaelsen, R.H.; Michie, D. and Boulanger, A. (1985) The Technology of Expert Systems, *Byte*, April, 303–12.

[11] Michalski, R.S. (1983) A Theory and Methodology of Inductive Learning, *Artificial Intelligence*, vol. 20, 111–61.

[12] Michalski, R.S. and Chilausky, R.L. (1980) Knowledge acquisition by encoding expert rules versus computer induction from examples: a case study involving soybean pathology, *International Journal of Man-Machine Studies*, vol. 12, 63–87.

[13] Milman, J.O. (1984) The Small Computer Revolution, *Proceedings of Compcon*, September, 20–6.

[14] Naylor, C. (1984) Discriminating Experts, *Practical Computing*, vol. 7(3), 108–12.

[15] Polanyi, M. (1967) *The Tacit Dimension* (New York: Anchor).

[16] Quinlan, J.R. (1979) Discovering Rules by Induction From Large Collections of Examples, in D. Michie (Ed.) *Expert Systems in the Microelectronic Age* (Edinburgh: Edinburgh University Press), 168–201.

[17] Quinlan, J.R. (1984) Learning Efficient Classification Procedures and Their Application to Chess End Games, in R.S. Michalski, J.G. Carbonell and T. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach* (Springer-Verlag), 463–82.

[18] Seidman, S. (1984) Putting the gloss on promising programs, *Computing* March 15, 36–37.

[19] Shanker, S.G. (1987) The Decline and Fall of the Mechanist Metaphor, in R. Born (Ed.) *The Case Against AI* (London: Croom Helm Ltd).
[20] Shapiro, A.D. (1983) The Role of Structured Induction in Expert Systems, unpublished Ph.D thesis, University of Edinburgh.
[21] Surya (1984) Expert-Ease, *Personal Computer World*, vol. 7(6), 208–13.
[22] Waterman, D.A. (1985) *A Guide to Expert Systems* (Reading, Mass,: Addison-Wesley Pub. Co.).