Myths and Legends in Learning Classification Rules

Wray Buntine*

Turing Institute
George House, 36 Nth. Hanover St.
Glasgow, G1 2AD, UK

Abstract

This paper is a discussion of machine learning theory on empirically learning classification rules. The paper proposes six myths in the machine learning community that address issues of bias, learning as search, computational learning theory, Occam's razor, "universal" learning algorithms, and interactive learning. Some of the problems raised are also addressed from a Bayesian perspective. The paper concludes by suggesting questions that machine learning researchers should be addressing both theoretically and experimentally.

Introduction

Machine learning addresses the computational problem of learning, whether it be for insight into the corresponding psychological process or for prospective commercial gain from knowledge learned. Empirical learning is sometimes intended to replace the manual elicitation of classification rules from a domain expert (Quinlan *et al.* 1987), as a knowledge acquisition sub-task for building classification systems. A classification rule is used to predict the class of a new example, where the class is some discrete variable of practical importance. For instance, an example might correspond to a patient described by attributes such as age, sex, and various measurements taken from a blood sample, and we want to predict a binary-valued class of whether the patient has an overactive thyroid gland. Empirical learning here would be the learning of the classification rule from a set of patient records.

The knowledge acquisition environment provides specific goals for and constraints on an empirical learning system: the system should fit neatly into some broader knowledge acquisition strategy, the system should be able to take advantage of any additional information over and above the examples, for instance, acquired interactively from an expert, the system should only require the use of readily available information, and of course the system should learn efficiently and as well as possible.

This paper proposes and discusses some myths in the machine learning community. All of these are twists on frameworks that have made significant contributions to our research, so the emphasis of the discussion is on qualifying the problems and suggesting solutions. The current flavour of machine learning research is first briefly reviewed before the so-called myths are introduced. The myths address the framework of bias, learning as search, computational learning theory, Occam's razor, the continuing quest for "universal" learning algorithms, and the notion of automatic non-interactive learning. While discussing these, a Bayesian perspective is also presented that addresses some of the issues raised. However, the arguments introducing the myths are intended to be independent of this Bayesian perspective. The conclusion raises some general questions for a theory of empirical learning.

Machine learning research

The development of learning systems in the machine learning community has been largely empirical and ideas-driven in nature, rather than theoretically motivated. That is, learning methods are developed based around good ideas, some with strong psychological support, and the methods are of course honed through experimental evaluation.

Such development is arguably the right approach in a relatively young area. It allows basic issues and problems to come to the fore, and basic techniques and methodologies to be developed. Although it should be more and more augmented with theory as the area progresses, especially where suitable theory is available from other sciences.

Early comments by Minsky and Papert (Minsky & Papert 1972) throw some light onto this kind of development approach. They were discussing history of research in the "perceptron" which is a simple linear thresholding unit that was a subject of intense study in early machine learning and pattern recognition.

They first comment on the attraction of the perceptron paradigm (Minsky & Papert 1972, page 18).

> Part of the attraction of the perceptron lies in the possibility of using very simple physical devices—"analogue computers"—to evaluate the linear threshold functions.
>
> ⋮

*Current address: RIACS, NASA Ames Res., MS 244-17, Moffet Field, CA 94035, (wray@ptolemy.arc.nasa.gov).

> The popularity of the perceptron as a model for an intelligent, general purpose learning machine has roots, we think, in an image of the brain itself ...

Good ideas and apparent plausibility were clearly the initial motivating force.

While perceptrons usually worked quite well on simple problems, their performance deteriorated rapidly on the more ambitious problems. Minsky and Papert sum up much of the research as follows (Minsky & Papert 1972, page 19):

> The results of these hundreds of projects and experiments were generally disappointing, and the explanations inconclusive.

It was only after this apparent lack of success that Minsky and Papert set about developing a more comprehensive theory of perceptrons and their capabilities. Their theory led to a more mature understanding of these systems from which improved approaches might have been developed. The lack of success, however, had already dampened research so the good ideas underlying the approach were virtually forgotten for another decade. Fortunately, a second wave of promising "neural net" research is now underway.

The major claim of this paper is that research on empirical learning within the machine learning community is at a similar juncture. Several promising frameworks have been developed for learning such as the bias framework (Mitchell 1980), the notion of learning as search (Simon & Lea 1974) and computational learning theory (Valiant 1985). It is argued in this paper that to progress we still need more directed theory to give us insight about designing learning algorithms.

A catalogue of myths and legends

The theoretical basis for bias research

If a learning system is to come up with any hypotheses at all, it will need to somehow make a choice based on information that is not logically present in the learning sample. The first clear enunciation of this problem in the machine learning community was by Mitchell, who referred to it as the problem of *bias* (Mitchell 1980). Utgoff developed this idea further, saying (Utgoff 1986, page 5)

> Given a set of training instances, *bias* is the set of all factors that collectively influence hypothesis selection. These factors include the definition of the space of hypotheses and definition of the algorithm that searches the space of concept descriptions.

Utgoff also introduced a number of terms: good bias is appropriate to learn the actual concept, strong bias restricts the search space considerably but independent of appropriateness, declarative bias is defined declaratively as opposed to procedurally, and preference bias is implemented as soft preferences rather than definite restrictions to the search space. Several researchers have since extended this theory by considering the strength of bias (Haussler 1988), declarative bias (Russell & Grosof 1987), the appropriateness of bias, and the learning of bias (Tcheng *et al.* 1989; Utgoff 1986). Much of this research has concentrated on domains without noise or uncertainty. In noisy domains, some researchers have considered the "bias towards simplicity", "overfitting", or the "accuracy vs. complexity tradeoff" (Fisher & Schlimmer 1988) first noticed in AI with decision tree learning algorithms (Cestnik *et al.* 1987).

There are, however, remaining open issues on this line of research. First, where does the original bias come from for a particular application? Are there domain independent biases that all learning systems should use? Researchers have managed to uncover through experimentation rough descriptions of useful biases, but a generative theory of bias has really only been presented for the case of a logical declarative bias. Second, investigation of bias in noisy or uncertain domains (where perfect classification is not possible) is fairly sparse, and the relation between the machine learning notion of bias and the decades of literature in statistics needs more attention. Third, there seems to be no separation between that component of bias existing due to computational limitations on the learner, bias input as knowledge to the original system, for instance defining a search goal, and therefore unaffected by computational limitations, and the interaction between bias and the sample itself.

What is required is a more precise definition of bias, its various functional components and how they can be pieced together so that we can generate or at least reason about a good bias for a particular application without having to resort to experimentation.

So while important early research identified the major problem in learning as bias, and mapped out some broad issues particularly concerning declarative bias, we have not since refined this to where we can reason about what makes a bias good. The first myth is that *there is currently a sufficient theoretical basis for understanding the problem of bias.*

Utgoff's definition of bias could, for instance, be decomposed into separate functional components.

Hypothesis space bias: This component of bias defines the space of hypotheses that are being searched, but not the manner of search.

Ideal search bias: This component of bias defines what a learning system should be searching for given the sample, that is, assuming infinite computing resources are available for search. As a contrast, one could consider algorithm bias as defining how the algorithm differs from the ideal.

Application-specific bias: This is the component of the bias that is determined by the application.

This decomposition is suggested by Bayesian techniques which give prescriptions for dealing with each

of these components (Buntine 1990). Bayesian techniques deal with belief in hypotheses, where belief is a form of preference bias. This gives, for instance, prescriptions for learning from positive-only examples, and for noisy or uncertain examples of different kinds.

For many learning systems described in the literature, ideal search bias is never actually specified, although notable exceptions exist (Quinlan & Rivest 1989; Muggleton 1987). Many publications describe an algorithm and results of the algorithm but never describe in general goal-oriented terms what the algorithm should be searching for, although they give the local heuristics used by the algorithm. Hence it is often difficult to determine the limitations of or assumptions underlying the algorithm. In areas such as constructive induction, where one is trying to construct new terms from existing terms used to describe examples, this becomes critical because the careful evaluation of potential new terms is needed for success. A survey of constructive induction methods (Matheus & Rendell 1989) reveals that few use a coherent evaluation strategy for new terms. This issue of search is the subject of the next myth.

Learning is a well-specified search problem

A seminal paper by Simon and Lea argued that learning should be cast as a search problem (Simon & Lea 1974). While few would disagree with this general idea, there appears to be no broad agreement in the machine learning community as to what precise goals a learning algorithm should be searching for. For a particular application, can we obtain a precise notion of the goal of search at all? Of course, the problems of bias and overfitting are just different perspectives of this same problem. The second myth, related to the first, is that *under the current view of learning as search, the goal of search is well-specified.* If it was generally known how to design a good bias then the search problem could be made well-specified.

Recurrent problems in machine learning such as splitting and pruning rules for decision trees (Mingers 1989) and the evaluation of new predicates for constructive induction (Matheus & Rendell 1989) are just some symptoms of this broad search problem.

There is one context, however, where learning as search is well-specified according to most current learning theories. Any reasonable model of learning or statistics has asymptotic properties that guarantee the model will converge on an optimal hypothesis. When a large enough quantity of data is available, it is easy to show the various statistical approaches become almost indistinguishable in result: maximum likelihood methods from classical statistics, uniform convergence and empirical risk minimisation techniques (Vapnik 1989) adopted by the computational learning community for handling logical, noisy and uncertain data, minimum encoding approaches (Wallace & Freeman 1987) and Bayesian methods (Buntine 1990). For instance, probably approximate correctness (PACness) is a notion used in computational learning theory to measure confidence in learning error (Valiant 1985). Under the usual definition of PACness, if confidence is high that error is low then the same will hold for a Bayesian method no matter what prior was used (this is a direct corollary of (Buntine 1990, Lemma 4.2.1)). In Bayesian statistics, "large enough" data means there is so much data that the result of a Bayesian method is virtually the same no matter what prior was used for the analysis; this situation is referred to as *stable estimation* (Berger 1985).

We say a sample is *sufficient* when the condition of "large enough" more or less holds for the sample. (Because the condition is about asymptotic convergence, it will only ever hold approximately.) This definition is about as precise as is needed for this paper. This means, for instance, that a sufficient sample is a reasonably complete specification of the semantics of the "true" concept but not its representational form. (Assume the classification being learned is time independent so the existence of a true concept is a reasonable assumption.)

A sufficient sample makes the search well-specified. For instance, if we are seeking an accurate classifier, then with a sufficient sample we can determine the "true" accuracy of all hypotheses in the search space reasonably well just by checking each one against the sample. That is, we just apply the principle of empirical risk minimisation (Vapnik 1989). With an insufficient sample, we can often check the accuracy of some hypotheses the same way—this is often the purpose of an independent test set—but the catch is we cannot check the accuracy of all hypotheses because inaccurate hypotheses can easily have high accuracy on the sample by chance. For instance, this happens with decision trees; it is well known they often have to be pruned because a fully grown tree has been "fitted to the noise" in the data (Cestnik *et al.* 1987).

A variation on this theme has been made by Weiss and colleagues who report a competitive method for learning probabilistic conjunctive rules (Weiss & Kapouleas 1989). They make the hypothesis space small (conjuncts of size three or less), so the sample size is nearly "sufficient". They then do a near exhaustive search of the space of hypotheses—something not often done in machine learning—to uncover the rule minimising empirical risk.

It is unfortunate, though, that a sufficient sample is not always available. We may only have a limited supply of data, as is often the case in medical or banking domains. In this case we would hope our learning algorithm will make the best use of the limited supply of data available. How can this be done?

Computational learning theory gives a basis for learning algorithms

Valiant's "theory of the learnable" was concerned with whether a machine can learn a particular class of con-

cepts in feasible computation (Valiant 1985). This theory centered around three notions: *uniform convergence* or distribution-free learning, that learning should converge regardless of the underlying distributions, *probable approximate correctness* (PACness), that the best a learner can do is probably be close to correct, and the need for *tractible algorithms* for learning. These three notions and variations have since been vigorously adopted for a wider range of learning problems by the theoretical community to create an area called computational learning theory.

The theory has been concentrated on the strength of bias and resultant worst-case complexity results about learning logical concepts. Evidence exists, however, that these results can be improved so better principles exist for the algorithm designer.

Simulations reported in (Buntine 1990) indicate a large gap exists between current worst-case error bounds obtained using uniform convergence and the kinds of errors that might occur in practice. The same gap did not occur when using a Bayesian method to estimate error, although the question of priors clouds these results. In addition, the current bounds only consider the size of the sample and not the contents of the sample. In the simulations, for instance, the space of consistent hypotheses sometimes reduced to one hypothesis quite quickly, indicating the "true" concept had been identified; yet bounds based on just the size of the sample cannot recognise this. This identification can occur approximately in that an algorithm may recognise parts of the concept have been identified. This behaviour has been captured using the "reliably probably almost always usefully" learning framework (Rivest & Sloan 1988), and a technique that generalises this same behaviour is implicit in a Bayesian method (Buntine 1990).

It is argued in (Buntine 1990) that these issues arise because of the worst-case properties of the standard PACness notion which is based on uniform convergence. While uniform convergence has desirable properties, it cannot be achieved when there is less data. In this context, less stringent principles give stronger guides. This is especially relevant in the context of learning noisy or uncertain concepts where a variety of other statistical principles could be used instead.

So the third myth is that *computational learning theory, in its current form, provides a solid basis on which the algorithm designer can perform his duties.* There are two important qualifications where this is not a myth. First, computational learning theory currently provides the algorithm designer with a skeletal theory of learning that gives a rough guide as to what to do and where further effort should be invested. Second, there are some areas where computational learning theory has provided significant support to the algorithm designer. For instance, in the previous section it was argued that with the notion of probably approximate correctness, computational learning theory provides a basis for learning in the context of a sufficient sample.

Occam's razor has a simple explanation

A standard explanation of Occam's razor (Blumer *et al.* 1987) can be summarized as follows (Dietterich 1989):

> The famous bias of Occam's Razor (prefer the simplest hypothesis consistent with the data) can thus be seen to have a mathematical basis. If we choose our simplicity ordering *before* examining the data, then a simple hypothesis that is consistent with the data is provably likely to be approximately correct. This is true regardless of the nature of the simplicity ordering, because no matter what the ordering, there are relatively few simple hypotheses.

An algorithm that looks for a simpler hypothesis under certain computational limitations has been called an Occam algorithm (Blumer *et al.* 1987). While the mathematical basis of Occam algorithms is solid, their useful application can be elusive. A poorly chosen Occam algorithm is rather like the drunk who, having lost his keys further down the road, only searches for them around the lamp post because that is the place where the light is strongest. Search should be directed more carefully.

Clearly, an Occam algorithm can only be said to provide *useful* support for Occam's razor if it will at least sometimes (not infrequently) find simpler hypotheses that are good approximations. The catch with Occam algorithms is that there is no guarantee they will do so. Suppose all reasonable approximations to the "true" concept are complex. Notice almost all hypotheses in a large space can be considered complex (since size is usually determined from the length of a non-redundant code). Then the framework of constructing a simplicity ordering and searching for simpler hypotheses has been pointless. If this almost always turns out to be the case, then the Occam algorithm approach will almost always not be useful. Certainly no proof has been presented that a guarantee of useful application exists; the main theorem in (Blumer *et al.* 1987) ignores this problem by *assuming* the "true" function is of size "at most n".

In other words, we need at least a weak guarantee that during learning, simpler hypotheses will sometimes be found that are good approximations. There are two potential arguments for this.

The first potential argument is that since we only require a good approximation, the hypothesis space can be reduced in size to one that is sufficient for finding a good approximation. A thorough treatment of this appears in (Amsterdam 1988). For the purposes of discussion, assume there is a uniform distribution on the examples and we are considering the space of all possible hypotheses defined over E kinds of examples. Such a space has size 2^E because each kind of example is either in the concept or not. Notice that for any one hypothesis, there are $2^{\epsilon E}$ hypotheses within error ϵ of it. So the smallest space of hypotheses guaranteed

to contain a hypothesis within ϵ of the "true" concept must be at least of size $2^{(1-\epsilon)E}$. So this first argument provides some support, but the reduction factor of just $(1-\epsilon)$ shows the argument is clearly not sufficient to provide a guarantee on its own.

Second, if we believe the simplicity ordering implicit in the Occam algorithm is somehow appropriate, that is, simpler hypotheses should be expected to be good approximations, then the Occam algorithm should be useful. Here, the power of the Occam algorithm comes from choosing a simplicity ordering appropriate for the problem in the first place. Bayesian methods support this principle because the simplicity ordering corresponds to prior belief. In minimum encoding approaches (Wallace & Freeman 1987) the principle is achieved by choosing an appropriate representation in which the "true" concept should be simple.

So the complexity argument above needs to be qualified: it is not useful in learning "regardless of the nature of the simplicity ordering". Either way, we need to think carefully about an appropriate simplicity ordering if we are to usefully employ the Occam algorithm.

The explanation for Occam's razor quoted above provides a mathematical basis for Occam's razor. However, the fourth myth is that *this mathematical basis provides a full and useful explanation of Occam's razor.* Two other supporting explanations have been presented that seem necessary to engage this mathematical basis. And we have not yet considered the case where concepts have noise or uncertainty. In this context different complementary arguments for Occam's razor become apparent (Wallace & Freeman 1987). Perhaps the key point is that Occam's razor finds practical application because of people's inherent ability to select key attributes and appropriate representations for expressing a learning problem in the first place (Michie 1986). There may also be other more subtle psychological explanations for its use.

"Universal" learning methods are best

Empirical learning seems to ignore one of the key lessons for AI in the 1970s called the strong knowledge principle (Waterman 1986, page 4):

> ... to make a program intelligent, provide it with lots of high quality specific knowledge about some problem area.

Whereas techniques such as explanation-based learning, analogical learning, and knowledge integration and refinement certainly embrace the strong knowledge principle, in empirical learning, as it is often described in the literature, one simply picks a universal learning method, inputs the data and then receive as output a classification rule. While early logical induction systems like Shapiro's MIS (Shapiro 1983) and subsequent similar systems do appear to incorporate background knowledge, they usually do so to extend the search space rather than to guide the search (Buntine 1988).

Some successful machine learning methodologies do incorporate weaker application-specific knowledge by some means. Two approaches are the careful selection of attributes in which examples are described (Quinlan *et al.* 1987; Michie 1986) and the use of various forms of interaction with an expert (Buntine & Stirling 1990). In addition, Bayesian statistics, with its notion of subjective knowledge or prior belief, could provide a means by which application-specific knowledge can be cautiously incorporated into the learning process.

Yet many comparative studies from the machine learning community, for instance, fail to consider even weak kinds of knowledge about an application that would help decide whether an algorithm is appropriate for an application, and hence whether the comparison of algorithms on the application is a fair one. Algorithms are applied universally to all problems without consideration of their applicability. A similar issue has been noted by Fisher and Schlimmer (Fisher & Schlimmer 1988, page 27).

> Using a statistical measure to characterize prediction tasks instantiates a methodology forwarded by Simon (1969) – domains must be characterized before an AI system's effectiveness can be properly evaluated. There is little benefit in stating that a system performs in a certain manner unless performance is tied to domain properties that predict system performance in other domains. Adherence to this strategy is relatively novel in machine learning.

The fifth myth is that *there exist universal learning algorithms that perform well on any application regardless.* Rather, there exist universal learning algorithms (and each of us provides living proof), but these can always be outperformed by a second class of algorithms better selected and modified for the particular application.

A way to develop this second class of non-universal learning algorithms is to develop "targeted" learning methods that, first, are suitable for specific kinds of classification tasks or specific inference models, and, second are able to be fine tuned or primed for the application at hand. The choices necessary in using these algorithms could be made in the light of subjective knowledge available, for instance, elicited in an interview with an expert. The correct choice of model is known to have considerable bearing on statistical problems like learning (Berger 1985, page 110).

At the broadest level we could choose to model the classification process with probabilistic decision trees or DNF rules, Bayesian or belief nets, linear classifiers, or perhaps even rules in a pseudo 1st-order logic such as DATALOG. And for each of these models there are many additional constraints or preferences that could be imposed on the representation to give the learning algorithm some of the flavour of the strong knowledge learning methods. One could choose to favour some

attributes over others in a tree algorithm or prime a belief net algorithm with potential causes and known independencies.

Learning should be non-interactive

One aspect of learning for knowledge acquisition, not sufficiently well highlighted in earlier statistical approaches, is the capability of promoting interaction with the expert to assist learning. This is done to obtain additional knowledge from the expert that may well be equivalent in value to a possibly expensive sample. The importance of interactive learning was recognised as early as 1947 by Alan Turing (Turing 1986, page 124), who said:

> No man adds very much to the body of knowledge, [*sic*] why should we expect more of a machine? Putting the same point differently, the machine must be allowed to have contact with human beings in order that it may adapt itself to their standards.

Interactive learning should be used with caution, however, because experts are often unreliable sources of knowledge. In the context of uncertainty, people have limitations with reasoning and in articulating their reasoning, so knowledge elicited must be interpreted with caution (Cleaves 1988). Also, we would hope that learning could still be achieved without interaction, perhaps at the expense of needing larger samples.

However, interaction is not always possible. Some applications require a so-called "autonomous agent". With these, not only should learning be entirely automatic, there may also be no-one present to help select an appropriate targeted learning algorithm as suggested in the previous section.

While most researchers now believe that learning can profit with careful expert interaction where possible, and research in this area exists (Angluin 1988; Buntine & Stirling 1990), the sixth myth, that *learning should be automatic and non-interactive*, lives on in many experimental studies reported and many of the algorithms under development.

Requirements for a theory of empirical learning

This section suggests questions that a theory for learning of classification rules should be addressing.

- How does the use of a learning system fit in a broader knowledge-acquisition strategy?
- According to what *inference model* should classification proceed, or in other words, what form of classification rules should be learned?
- What is the *induction protocol* or typical course of an induction session? What sorts of questions can the trainer reasonably answer, and with what sort of knowledge can the trainer prime the learning system?
- For a particular induction protocol and inference model, how should the system perform induction given its computational resources? What is being searched for, and how should this search be performed?
- What are the average and worst-case computational and data requirements of the system for a given problem? Furthermore, what problems can be solved with reasonable computational resources, and what amounts of data should be needed?
- How does a theory of uncertainty relate to the problem of learning classification rules? How can this then throw light on the previous search problem?
- When and how should subjective knowledge, weak or strong domain knowledge, or other information extraneous to the sample be incorporated into the learning process? If this is done but poorly, how can the system subsequently detect from evidence in the sample that the incorporated subjective knowledge is actually inappropriate to the application?
- How reliable are the classification rules learned by the system from available data? How much more data is required, and of which type? Can the system ask a few pertinent questions or design a few key experiments to improve subsequent results? Are the results sensitive to assumptions implicit in the system? (In Bayesian methods, this includes priors.)

It has been argued at various places throughout this paper that Bayesian theory can address at least some of these questions. It is certainly a (seventh) myth, however, that Bayesian methods provide a complete theory for designing learning algorithms. There are many complementary statistical perspectives and many complementary theoretical tools such as optimisation and search, decision theory, resource-bounded reasoning, computational complexity, models of man-machine interaction, and the psychology of learning, etc. In addition, some of the above questions require a pragmatic and experimental perspective, particular those concerned with the human interface and learning methodology.

Acknowledgements

Brian Ripley, Donald Michie and RIACS gave support and Peter Cheeseman, Pete Clark, David Haussler, Phil Laird, Steve Muggleton, Tim Niblett, and the AAAI reviewers provided constructive feedback. Any remaining errors are my own.

References

Amsterdam, J. 1988. Extending the Valiant learning model. In *Fifth International Conference on Machine Learning*, 381–394, Ann Arbor, Michigan. Morgan Kaufmann.

Angluin, D. 1988. Queries and concept learning. *Machine Learning*, 2(4):319–343.

Berger, J. O. 1985. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M.K. 1987. Occam's razor. *Information Processing Letters*, 24:377–380.

Buntine, W.L. and Stirling, D.A. 1990. Interactive induction. In Hayes, J.; Michie, D.; and Tyugu, E., editors, *MI-12: Machine Intelligence 12, Machine Analysis and Synthesis of Knowledge.* Oxford University Press, Oxford.

Buntine, W.L. 1988. Generalised subsumption and its applications to induction and redundancy. *Artificial Intelligence*, 36(2):149–176.

Buntine, W.L. 1990. *A Theory of Learning Classification Rules.* PhD thesis, University of Technology, Sydney. Forthcoming.

Cestnik, B.; Kononenko, I.; and Bratko, I. 1987. Assistant 86: A knowledge-elicitation tool for sophisticated users. In Bratko, I. and Lavrač, N., editors, *Progress in Machine Learning: Proceedings of EWSL-87*, 31–45, Bled, Yugoslavia. Sigma Press.

Cleaves, D.A. 1988. Cognitive biases and corrective techniques: proposals for improving elicitation procedures for knowledge-based systems. In Gaines, B. and Boose, J., editors, *Knowledge Acquisition for Knowledge-Based Systems*, 23–34. Academic Press, London.

Dietterich, T.G. 1989. Machine learning, Tech. Report 89-30-6, Dept. Comp. Sc., Oregon State University. To appear, *Annual Review of Computer Science*, Vol. 4, Spring 1990.

Fisher, D.H. and Schlimmer, J.C. 1988. Concept simplification and prediction accuracy. In *Fifth International Conference on Machine Learning*, 22–28, Ann Arbor, Michigan. Morgan Kaufmann.

Haussler, D. 1988. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36(2):177–222.

Matheus, C.J. and Rendell, L.A. 1989. Constructive induction on decision trees. In *International Joint Conference on Artificial Intelligence*, 645–650, Detroit. Morgan Kaufmann.

Michie, D. 1986. The superarticulacy phenomenon in the context of software manufacture. *Proc. Roy. Soc. (A)*, 405:185–212.

Mingers, J. 1989. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3(4):319–342.

Minsky, M. and Papert, S. 1972. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, second edition.

Mitchell, T.M. 1980. The need for biases in learning generalisations, CBM-TR 5-110, Rutgers University, New Brunswick, NJ.

Muggleton, S.H. 1987. Duce, an oracle based approach to constructive induction. In *International Joint Conference on Artificial Intelligence*, 287–292, Milan.

Quinlan, J.R. and Rivest, R.L. 1989. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248.

Quinlan, J.R.; Compton, P.J.; Horn, K.A.; and Lazarus, L. 1987. Inductive knowledge acquisition: A case study. In Quinlan, J.R., editor, *Applications of Expert Systems.* Addison Wesley, London.

Rivest, R.L. and Sloan, R. 1988. Learning complicated concepts reliably and usefully (extended abstract). In *Seventh National Conference on Artificial Intelligence*, 635–640, Saint Paul, Minnesota.

Russell, S.J. and Grosof, B.N. 1987. A declarative approach to bias in concept learning. In *Sixth National Conference on Artificial Intelligence*, 505–510, Seattle.

Shapiro, E.Y. 1983. *Algorithmic Program Debugging.* MIT Press.

Simon, H.A. and Lea, G. 1974. Knowledge and cognition. In Gregg, L.W., editor, *Knowledge and Cognition*, 105–127. Lawrence Erlbaum Associates.

Tcheng, D.; Lambert, B.; Lu, S. C-Y.; and Rendell, L. 1989. Building robust learning systems by combining induction and optimization. In *International Joint Conference on Artificial Intelligence*, 806–812, Detroit. Morgan Kaufmann.

Turing, A.M. 1986. Lecture to the London Mathematical Society on 2 February 1947. In Carpenter, B.E. and Doran, R.W., editors, *AM Turing's Ace Report of 1946 and other papers*, 106–124. MIT Press.

Utgoff, P.E. 1986. *Machine Learning of Inductive Bias.* Kluwer Academic Publishers.

Valiant, L.G. 1985. A theory of the learnable. *CACM*, 27(11):1134–1142.

Vapnik, V.N. 1989. Inductive principles of the search for empirical dependencies. In Rivest, R.; Haussler, D.; and Warmuth, M.K., editors, *COLT'89: Second Workshop on Computational Learning Theory*, 3–21, University of California, Santa Cruz. Morgan Kaufmann.

Wallace, C.S. and Freeman, P.R. 1987. Estimation and inference by compact encoding. *J. Roy. Statist. Soc. B*, 49(3):240–265.

Waterman, D.A. 1986. *A Guide to Expert Systems.* Addison Wesley.

Weiss, S.M. and Kapouleas, I. 1989. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *International Joint Conference on Artificial Intelligence*, 781–787, Detroit. Morgan Kaufmann.