In: Machine and Human Learning, pp35-49, Eds: Y. Kodratoff and A. Hutchison, London: Kogan Page, (1989). http://www.cs.utexas.edu/users/pclark/papers

Knowledge Representation in Machine Learning

Peter Clark
The Turing Institute
36 N.Hanover St
Glasgow, UK

1989

1 Introduction

Knowledge representation is a topic poorly discussed in machine learning. However, it is perhaps the fundamental consideration in the design of any learning system, because the representation used determines to a great degree what can and cannot be learned.

This chapter presents a survey of different schemes used to represent learned knowledge in machine learning systems. Firstly, a brief description is given of the task of designing an adaptive system and of the central role which the method of representing initial and learned knowledge plays. Secondly we examine different representational schemes which have been used in learning systems, discussing the motivation for them, the way in which they are used and their limitations.

The reader should note that we do not discuss representations appropriate for connectionist or genetic approaches to learning. This is not to dismiss them as less important, but merely to constrain the scope of this chapter to a tractable size.

2 The Learning Task

2.1 Learning and World Modelling

An adaptive system must have, by definition, the capacity to perform a task in more than one way. Consequently the system must make choices about the most appropriate course of action to take, and if the system is adaptive it should be able to modify its choice-making behaviour should a choice turn out to be inappropriate.

In order to make choices it is necessary to predict the likely outcomes of making them, requiring some kind of internal model of the world. An adaptive system can be viewed as performing two processes involving this model: firstly, to use it to respond most appropriately to the environment, and secondly to continuously extend and correct such a model in the light of success or failure, keeping it up-to-date and efficient.

Knowledge representation concerns the issue of how we can best create, maintain and use such a model of the world. Learning systems in particular must concern themselves with issues of:

1. What sort of errors might occur in the representation

2. How to detect, locate and correct them

This view of learning as a modelling process is sometimes useful and one to which we shall occasionally refer in this chapter.

2.2 Learning and Problem-Solving

Before embarking on a survey of representation schemes, a few words should be given as to the relationship between 'learning' and 'non-learning' systems. Learning is often loosely described as the improvement of performance with experience – although this is a rather general definition, a more precise definition is notoriously difficult to pin down. Rich [45] suggests that the reason for this is because two key components of learning, namely

- the acquisition of knowledge
- the integration of knowledge into a system

are also important to *other* tasks which are not normally considered as learning tasks. For example, many expert systems perform simple data acquisition but are not normally referred to as 'learning', and many 'non-learning' diagnostic systems search to integrate new observations with expert-supplied knowledge of the domain to produce a diagnosis. There is certainly a close correspondence between learning and problem-solving. Indeed, a useful way of viewing learning is as a special kind of problem-solving task – one in which the problem to be solved is that of self-diagnosis and improvement rather than the diagnosis of something external to the system. Thus it follows that many of the problem-solving techniques used in 'non-learning' systems are also important in learning systems, and that developed methods of knowledge representation to facilitate problem-solving in these systems are also relevant to learning systems.

2.3 Learning and Knowledge Representation

It is also worth briefly considering the functions which a representation of knowledge must perform. Konolige [22] suggests it is useful to view each component of a representational system as having two important roles:

1. **A 'semantic' role:** The component has a meaning, denoting some object, relationship or event in the world.
2. **A 'computational' role:** The *syntactic* structure of the component, combined with the system's inference machinery, determines the ways in which the knowledge is to be used.

For learning systems, this 'computational' role played by the syntactic structure of a component of representation is a dual one:

- As 'program' to perform the application task for which the system has been designed
- As 'data' to be manipulated by the learning component of the system

Thus for learning systems the syntactic nature of a representation has roles in both performing a task and facilitating learning. These have an influence in determining which representational scheme will be effective in a learning system: the syntax of represented knowledge must interact appropriately not only with the inference machinery during a performance task but also with the machinery for learning.

As an example of the important role of syntax, consider expressing some simple fact, e.g. "the diode is broken", in logic. While it may seem arbitrary to either represent this as `broken(diode)` or `state(diode,broken)`, the two statements may have different influences on the system's ability to learn. For instance, generalisation is often defined in terms of syntactic operations (e.g. replace constants with variables), and thus the sets of possible generalisations which the system can generate of `broken(diode)` and `state(diode,broken)` may differ. To ensure proper performance and learning, the correspondence between desired and actual computational role in the system's learning machinery must be maintained. This correspondence is cited as one of the reasons for the success of the learning system AM [25].

3 Survey of Representational Systems

3.1 Rule Induction from Examples

Learning can be seen as a task of creating and maintaining an internal model of the world. In its most general form this is a task of immense proportions, but fortunately there are simpler variants. One of the most constrained variants is the task of learning, from a set of training examples, to predict a particular unknown property or 'class' of a situation from a set of known properties of that situation, making a number of simplifying assumptions which we describe below (an example of this is learning to make a medical diagnosis given a set of patient's symptoms). This is sometimes referred to as 'rule induction from examples'. Despite its constrained nature, it remains the most successful of learning paradigms in commercial applications [36] and substantial research effort in the machine learning community has been devoted to it.

In its most constrained form, concept learning systems assume that the class to predict is solely a function of the properties or 'attributes' which the programmer has used to describe the training examples. There is no dependence on any previously seen situations (no 'state variables') and no dependence of any other knowledge with which the concept must be integrated. Secondly, as these systems operate by conducting a heuristic search of the space of concepts, the concept representation method must have the properties of allowing such a search to be both easily definable and sufficiently constrained to locate predictive rules with available search heuristics. To meet these requirements, the concept description is usually assumed to be a boolean function of tests on the individual attributes describing examples. As a result, a search based on a general-to-specific method is easily definable. ID3 [41], AQ11 [26] and CN2 [5] are examples of such systems, differing in the way the boolean combination of tests is built up (as a decision tree, an unordered set of rules, and an ordered list of rules respectively). The system Lex [34] constrained search even further by permitting only conjuncts of tests in the concept description, thus allowing an exhaustive rather than heuristic search of the concept description space to be computationally feasible. Other variants include PROMIS [21], making a bi-directional exploration of the search space.

Despite their usefulness, concept learning systems such as ID3 and AQ11 have limitations. In the rest of this chapter, we examine these and what changes in representational schemes are used to overcome them.

3.2 First Order Logic Representations

3.2.1 Increasing Representational Power

Concept learning systems operate under two constraining assumptions:

- The expressive power of the concept representation language is limited
- They assume no other knowledge to be present in the system with which the concept descriptions must be integrated

There are good reasons for making these assumptions – to constrain and simplify the search respectively. To relax these assumptions, the knowledge representation must both become more expressive while still coupling effectively with efficient search strategies.

The concept learning systems described earlier use a propositional-like logic based solely on conjuncts and disjuncts of attribute tests. One way of improving the expressive power of the knowledge representation used is to move to a more powerful logic such as a first order logic. A large number of machine learning systems using representational languages more powerful than propositional logic employ some type of first order logic to represent learned knowledge. Two classes of such representations are particularly common: those based on Horn clause logic, and those based on 'production rules' (in a 'pure' form, i.e. without measures of certainty or arbitrary procedures attached). Logics based on Horn clauses constrain clauses to have at most one positive literal, and form the basis of several logic programming languages including Prolog. Production rule representations can be viewed as equivalent to a first order logic representation in which the inference machinery is *incomplete* (there are some true facts which cannot be derived, e.g. given `if A then B` and `not B`, the system cannot deduce `not A`). Konolige provides an analysis of this class of system in [22]. As these two classes of representation are closely related, we treat them together in this section. In [49], Sammut presents a survey of the use of logic in machine learning. Mozetic and Lavrac [38] present an analysis of how increasing the expressive power of a representation degrades its efficiency in learning.

3.2.2 Constraining Search

Marvin [50] and MIS [53] both learn logic programs from examples. Given this more powerful representation, extra techniques for controlling the search are needed. Both these systems make use of an *oracle*, namely an infallible source of knowledge which can answer 'yes' or 'no' to any query the system puts to it. Both still assume that concepts can be described concisely in terms of the previous attributes and concepts the system has encountered. A final powerful constraint is the assumption that there is no noise, and hence counterexamples to a hypothesised concept necessarily mean the hypothesis is incorrect. These systems are particularly notable for their carefully formalised searches of the space of programs to learn, and their methods for keeping the number of queries to the oracle small.

3.2.3 Introducing New Terms

Cigol [39] is another system learning logic programs from examples, based on inverting the resolution mechanism used in deductive inference. It again uses an oracle and assumes no noise, but is capable of introducing 'new terms' (i.e. intermediate functions of known attributes) which enable otherwise unwieldy concept descriptions to become expressible concisely. An analysis of the importance of constructive induction in the learning of certain classes of concepts is made by Rendell [44]. Such automatic introduction of terms is sometimes referred to as 'constructive induction'.

A simple example of the importance of introducing intermediate terms can be seen by considering the problem of learning to recognise 'checkmate' in a chess end-game. While in theory ID3 can learn this given only the x and y co-ordinates of the chess pieces, in practice a very large number of examples are required and the concept description would be highly complex. However, given the introduction of simple intermediate terms based on the initial attributes (e.g. "on same rank or file", if $x_1 = x_2$ or $y_1 = y_2$) the concept representation becomes considerably simpler and easier to learn. Constrained resources mean only a few simple concept descriptions can be searched – thus introducing terms likely to simplify the target concept description will improve learning. The importance of structuring problems in this way has also been investigated by Shapiro and Niblett [52].

3.2.4 Improving Efficiency

In addition to learning 'new' knowledge (knowledge which does not follow deductively from what is already known), some work in machine learning has been devoted to making existing knowledge more efficient. The area of explanation-based learning (EBL) covers a variety of techniques, most working with some first order logic variant such as Horn clauses or production rules as a representational method. The most common technique, explanation-based generalisation (EBG), involves generating and storing a general solution to a problem [33, 19]. Solving a problem often involves instantiating and applying a number of operators or 'rules' – explanation-based 'generalisation' ('re-generalisation' is perhaps more appropriate) collects and simplifies the uninstantiated operator sequence, storing it for later use. EBG can in fact be viewed as selectively applying the logic programming method of partial evaluation [55].

EBG improves efficiency when the cost of generating, storing and retrieving these learned operators is outweighed by the improved speed they give (a learned solution sequence no longer needs to be recalculated and instead can be immediately applied). Early work assumed generalising solutions to old problems would yield such an overall benefit (thus assuming new problems will be similar to old problems). However, this is not necessarily the case – Minton [31, 30] has recently conducted more detailed analyses of the criteria for deciding which generalisations to store in order to achieve an increase in the system's performance.

3.3 Integrating New Knowledge

We have described several methods by which languages based on first order logic are used to represent learned knowledge. Increasing the representational power in this way requires the use of extra techniques for constraining search, one of which is the introduction of an oracle to ensure that at each stage of the learning process the system has added correct knowledge to its knowledge base.

In the absence of an oracle, it may happen that erroneous information is added and not detected as being in error until a later stage. As a result, computations dependent on that data may also be in error. Truth maintenance systems (e.g. by Doyle [9] and deKleer's ATMS [8]) offer methods by which statements in logic can be tagged with information about their dependencies. The advantages of these representations, particularly in ATMS systems, are two-fold. Firstly, the original source(s) of an inconsistency can immediately be located, and secondly conclusions derived solely from the source(s) can be easily identified and removed. Truth maintenance approaches are based on the more general principle augmenting a representation with 'meta-knowledge' expressing the basis for considering a learned fact to be true. This knowledge is extremely useful for maintaining an expanding base of knowledge, and is thus an important representational method for machine learning. Truth maintenance can be viewed as a computational method of reasoning with normal rules in default logic [43].

There are also other approaches besides truth maintenance for managing the integration of new knowledge into a system. While a truth maintenance system can only alter its knowledge by changing the truth values of rules and facts in its knowledge base, other systems allow rules to be modified and/or new rules added. Some systems (e.g. [42, 57]) will allow a rule's <condition> part to be specialised if it is in error by adding extra conditions or adding notes of exceptions or 'censors' to the rule. Others (e.g. [56, 15]) will conduct a search for new rules to add to the system to achieve correct behaviour. This is a complex search problem as there are often many alternative solutions to select between, and indeed is a similar task to that performed by expert systems searching for a 'best' interpretation of a set of data.

3.4 Representing Uncertainty

As discussed in Section 2.3, knowledge represented in a learning system has a dual role – to be reasoned *about* during learning, and reasoned *with* during a performance task. While learning systems must deal with uncertainty during learning (induction is inherently uncertain), the output of that learning in the systems discussed so far are for use within a framework of certain reasoning, where there is no representation of uncertain or probabilistic knowledge. This is not necessarily always desirable. We may wish the application of the learned knowledge to handle uncertainty in some way, for example by providing a probability of correctness in a classification task, or using background knowledge which is not known with absolute certainty.

Fortunately the systems for concept learning from examples (described in Section 3.1) can be extended to produce some measure of probability in classification (e.g. AQ15 [27] and in ID3 [40]). However when learning is to occur in the presence of *uncertain* background knowledge, the problems become more difficult. Surprisingly (surprisingly because most AI applications use some method for representing uncertainty) little research into learning when uncertain background knowledge is available has been conducted.

It is worth mentioning one of the reasons why including a representation of uncertainty in learned knowledge is difficult. A fundamental problem with uncertain inference (or indeed any non-monotonic reasoning system) is that the dependence between components of what is known ceases to be tightly constrained. For example, if I know `A` $\Rightarrow$ `B` and `A`, then I can conclude `B` and know that `B` is true no matter what other inferences I make. The truth of `B` depends solely on the two premises. However if I know `A` $\Rightarrow$ `B with probability 50%` and `A`, what can I conclude about `B`? Here the probability of `B` depends not only on these two premises but also on everything else relevant to `B` which is known (and not known). Learning involves being able to identify and correct faulty components in a model of the world. To do this, some knowledge of what the components are and how they interact is necessary. Representations of uncertainty can result in highly complex interactions between many components of a system, and are thus difficult to handle in the learning module of a system.

Probabilities represent one type of uncertainty, namely knowledge of imperfect correlations in some population of situations or events. However, there are other types of uncertainty besides this, for example where it is known some (perfect) correlation exists between properties in a domain but it is not known exactly what form that correlation takes (e.g. I know there is a correlation between the country I wish to visit and the need to get a visa, but I don't know what that correlation is). This notion of uncertain knowledge about (perfect) functional dependencies has been formalised by Davies and Russell in their theory of determinations [7], and used for helping to constrain inductive learning [46, 47]. A specialised formalisation of this type of uncertain knowledge for representing 'arguments' for and against a hypothesis has also been used for assisting in learning tasks [6].

3.5 Exemplar-Based Representations

There has recently been an interest in exemplar-based representations for machine learning (e.g. the systems Protos [1], Nexus [2], and by Kibler and Aha [20]). A clear definition of 'exemplar-based representation' is hard to formulate, and the phrase refers as much to a learning methodology as to a new representational scheme. Exemplar-based systems are often characterised by the following properties:

- Complete descriptions of selected examples are stored in memory
- For a classification task, a matching algorithm locates the stored exemplar most similar to the example to be classified
- The classification of the retrieved exemplar is assigned to the new example

This methodology can be viewed approximately as one of deferring generalisation until a run-time classification task is at hand (rather than enumerating generalisations beforehand as AQ11, say, would do). The matching algorithm itself implicitly defines how to generalise from known exemplars. In general the 'bias' (i.e. criteria for which generalisations to prefer, given a choice) in exemplar-based approaches differs from that of rule learning systems (see [4]), and some authors have argued this bias is more appropriate for 'natural' concepts (e.g. [1]). There are corresponding computational resource trade-offs, and also important similarities, between this methodology and that of learning classification rules. This is discussed in more detail in [4].

3.6 Structured Representations of Knowledge

3.6.1 Introduction

There are several methods for representing larger 'structures' of knowledge by which known facts can be grouped and organised together. Examples of these are frames [29] and scripts [51]. Recall in Section 2.3 that a representation of knowledge has two roles – a semantic role in denoting some object or event, and a computational role in helping determine how that knowledge should be used. While structured representations like frames and scripts may not provide greater expressive power (indeed Hayes [17] argues that knowledge represented by frames can easily be re-expressed in logic), the *computational* role they play can be valuable.

Many 'non-learning' expert systems perform the task of forming a plausible model to account for observations, based on the knowledge they have. However, this is precisely the task of a learning system too, and hence techniques of search and knowledge representation used in these 'non-learning' systems are relevant. In particular, expert systems often form hypotheses about observations at a number of different levels of detail (e.g. the systems Internist [28], Hearsay [10] and Abstrips [48]), and Clancey [3] provides an analysis of this general framework. This structured representation is useful for efficient search (although it is difficult to control) as whole classes of hypotheses can be quickly and approximately checked to decide whether a more detailed analysis of them is merited.

Despite the importance of structured representations in 'non-learning' expert systems, there has been limited research into their role for machine learning. One reason for this is that it is difficult to formulate methods for analysing and understanding the behaviour of more complex representational schemes, a prerequisite for proper design of a system and the drawing of general conclusions from experimental results. Frames were used as the basic knowledge representation scheme in the learning system AM [24], but have been little used since then in machine learning systems. Mozetic [37] has performed work in the use of abstraction hierarchies as a representation to assist in learning qualitative models of the heart. The system Genesis [35] uses a hierarchical organisation of 'schemas' to find the ones best matching a given piece of narrative, and will augment and alter such schemas to account for details in the narrative, sometimes generalising the resulting new or modified schema to add to its store. Again, schemas as a structured set of facts play an important computational role in determining what to expect in a situation, and their hierarchical organisation is computationally important in the search for a best matching schema.

3.6.2 Reasoning by Analogy

One type of learning system where structured knowledge is used is that for analogical reasoning. Reasoning by analogy is again difficult to define precisely (Gentner [13] argues that its distinction from other forms of learning is more of a continuum rather than a clear divide), but can be loosely described as the task of finding similarities between two separate systems of knowledge. This idea has been developed in Gentner's structure mapping theory [14] and implemented in the Structure Mapping Engine [12]. Analogy systems based on this idea require some representation of a 'structured system of knowledge', most simply achieved by grouping a set of facts and relations together (e.g. by tagging them with a label as to the system they belong to). This approach has been used in several analogy systems (eg. [11, 18]).

3.6.3 Retaining Structured Solution Methods

We mentioned earlier that systems using explanation-based generalisation retained (generalised) solution methods for future use. While this technique is used in a deductive framework of reasoning, other researchers have advocated the more general approach of retaining previous solution methods not only for improving efficiency but also for assisting in solving new problems where they cannot be directly applied. Instead, modifications must be made to the old solution method before it can be used.

This approach has been particularly used in the domain of planning. The system Hacker [54] used a 'library' of plans to solve problems, learning appropriate plan modification rules as the system was run. Chef [16] uses and extends a similar store of old plans in order to solve new planning problems. These approaches again require a structure for representing a 'solution method' (eg. a sequence of operators to apply) to be represented.

3.7 Procedural Representations

Much work in machine learning deals with 'declarative' problems (i.e. not involving notions of sequences of events) and this is reflected in the emphasis on such problems in the preceding sections. However in some domains, (e.g. game playing, robot planning), the required solution is an ordered set of actions rather than some 'fact'.

Two things have enabled work on 'declarative' problems to be also applied to problems of planning and ordering. Firstly, an important result of work on planning systems in the early 1970's was to show the correspondence between finding a plan to solve a goal, and finding a proof for a theorem. As a result, methods used in 'declarative' problems can also be applied to planning problems, with the modification that the solution is the proof of a fact rather than the fact itself. Work on explanation-based learning (Section 3.2) has partly concentrated on planning tasks (e.g. [23, 32]) using this correspondence between plans and proofs as a basis for finding solutions. Secondly, the task of deciding an action to perform can be viewed as a classification problem, and hence concept learning systems are also applicable to the tasks of control and planning. For example, the concept learning module of Lex [34] learns to classify integration operators into 'appropriate to perform in the current state' and 'inappropriate'.

4 Summary

Knowledge representation is a crucial issue to address in the design of any system, including those desired to learn. We have argued in Section 2 that

- Learning can be viewed as a particular type of problem-solving, and thus search and representation schemes used in 'non-learning' systems are also of relevance to machine learning
- That a representational scheme used in a learning system plays a computational (as well as a semantic) role in determining how knowledge is used

Some findings from our survey are as follows:

- Rule induction systems use simple propositional-like logic representations, enabling search to be adequately constrained
- Such learning systems can easily augment such rules to include measures of certainty or probability
- Most other research in machine learning uses representations based on variants of first order logic, especially production rule representations and Horn clause logic
- Increasing the representational power in this way and including background knowledge requires additional search techniques to make learning possible
- Little research, apart from that on rule induction systems, has been conducted into learning with representations of uncertain knowledge
- A small amount of research in machine learning has been conducted into using other techniques of knowledge representation, including truth maintenance approaches, exemplar-based reasoning and using multiple levels of abstraction.

References

[1] Bareiss, E. R., Porter, B. W., and Wier, C. C. (1987). PROTOS: an exemplar-based learning apprentice. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann.

[2] Bradshaw, G. (1987). Learning about speech sounds: the nexus project. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann, pp. 1–11.

[3] Clancey, W. J. (1984). Classification problem solving. In *AAAI-84* , pp. 49–55.

[4] Clark, P. (1988). A comparison of exemplar-based and rule-based learning systems. In *International workshop on Machine Learning, Meta-reasoning and Logics* Portugal, Portugal, Ed., Faculdade de Economia, Univ. Porto, pp. 69–81. (Also available as TIMLG-15, Turing Inst., Glasgow, UK).

[5] Clark, P., and Niblett, T. (1989). The cn2 induction algorithm. *Machine Learning Journal Vol 3*. (In press).

[6] Clark, P. E. (1988). Representing arguments as background knowledge for constraining generalisation. In *Proc. Third European Working Session on Learning (EWSL-88)* London, D. Sleeman, Ed., Pitman, pp. 37–44.

[7] Davies, T. R., and Russell, S. J. (1987). A logical approach to reasoning by analogy. In *IJCAI-87* , pp. 264–270.

[8] DeKleer, J. (1984). Choices without backtracking. In *AAAI–84* .

[9] Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence 12*, 231–272.

[10] Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. (1981). The hearsay-II speech-understanding system: integrating knowledge and resolving uncertainty. In *Readings in Artificial Intelligence*, B. L. Webber and N. J. Nilsson, Eds., Tioga, Ca, pp. 349–389.

[11] Falkenhainer, B. (1987). Scientific theory formation through analogical inference. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann, pp. 218–229.

[12] Falkenhainer, B., Forbus, K. D., and Gentner, D. (1986). The structure-mapping engine. In *AAAI-86* , pp. 272–277.

[13] Gentner, D. (1988). Analogical inference and analogical access. In *Analogica*, A. Prieditis, Ed., Kaufmann, Ca, pp. 63–88.

[14] Gentner, D. (1983). Structure-mapping: a theoretical framework for analogy. *Cognitive Science 7*, 2, 155–170.

[15] Hall, R. (1986). *Learning by Failing to Explain.* Technical Report 906 (AI-TR-906), MIT AI Laboratory.

[16] Hammond, K. (1986). CHEF: a model of case-based planning. In *AAAI-86* .

[17] Hayes, P. J. (1985). The logic of frames. In *Readings in Knowledge Representation*, R. J. Brachman and H. J. Levesque, Eds., Kaufmann, Ca, pp. 287–295.

[18] Keane, M., and Brayshaw, M. (1988). The incremental analogy machine: a computational model of analogy. In *Proc. Third European Working Session on Learning (EWSL-88)* London, D. Sleeman, Ed., Pitman, pp. 53–62.

[19] Kedar-Cabelli, S. T., and McCarty, L. T. (1987). Explanation-based generalization as resolution theorem proving. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann, pp. 383–389.

[20] Kibler, D., and Aha, D. W. (1987). Learning representative exemplars of concepts: an initial case study. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann.

[21] King, R. (1987). An inductive learning approach to the problem of predicting a protein's secondary structure from its amino acid sequence. In *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, I. Bratko and N. Lavrač, Eds., Sigma, Wilmslow, UK.

[22] Konolige, K. (1986). *A Deductive Model of Belief.* Pitman, London.

[23] Laird, J. E., Rosenbloom, P. S., and Newell, A. (1986). Chunking in soar: the anatomy of a general learning mechanism. *Machine Learning 1*, 1, 11–46.

[24] Lenat, D. (1983). The role of heuristics in learning by discovery: three case studies. In *Machine Learning, vol. 1*, J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, Eds., Tioga, Palo Alto, Ca.

[25] Lenat, D. B., and Brown, J. S. (1984). Why am and eurisko appear to work. *Artificial Intelligence 23*, 3, 269–294.

[26] Michalski, R. S., and Larson, J. (1983). *Incremental Generation of VL_1 Hypotheses: the underlying Methodology and the Description of Program AQ11.* ISG 83-5, Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, Urbana.

[27] Michalski, R. S., Mozetic, I., Hong, J., and Lavrac, N. (1986). The AQ15 inductive learning system : an overview and experiments. In *Proceedings of IMAL 1986* Orsay, France, Université de Paris-Sud.

[28] Miller, A. M., Pople, H. E., and Myers, J. D. (1982). Internist-i, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine 307*, 468–476.

[29] Minsky, M. (1985). A framework for representing knowledge. In *Readings in Knowledge Representation*, R. Brachman and H.Levesque, Eds., Kaufmann, California.

[30] Minton, S. (1988). Quantitative results concerning the utility of explanation-based learning. In *AAAI-88* Ca, Kaufman, pp. 564–569.

[31] Minton, S. (1985). Selectively generalizing plans for problem-solving. In *IJCAI-85* , pp. 596–599.

[32] Minton, S., Carbonell, J. G., Etzioni, O., Knoblock, C. A., and Kuokka, D. R. (1987). Acquiring effective search control rules: explanation-based learning in the prodigy system. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann, pp. 122–133.

[33] Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986). Explanation-based generalization: a unifying view. *Machine Learning Journal 1*, 1, 47–80.

[34] Mitchell, T. M., Utgoff, P., and Banerji, R. (1983). Learning by experimentation: acquiring and refining problem-solving heuristics. In *Machine Learning, vol. 1*, J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, Eds., Tioga, Palo Alto, Ca.

[35] Mooney, R., and DeJong, G. (1985). Learning schemata for natural language processing. In *IJCAI-85* , pp. 681–687.

[36] Mowforth, P. (1986). *Some Applications with Inductive Expert System Shells.* TIOP 86-002, Turing Institute, Glasgow, UK.

[37] Mozetic, I. (1987). The role of abstractions in learning qualitative models. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann.

[38] Mozetic, I., and Lavrac, N. (1988). Incremental learning from examples in a logic-based formalism. In *International workshop on Machine Learning, Meta-reasoning and Logics* Portugal, P. Brazdil, Ed., Faculdade de Economia, Univ. Porto, pp. 109–127. (Proceedings to be published in book form in 1989).

[39] Muggleton, S., and Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. In *Proc. 5th Int. Conf. on Machine Learning* Ca, J. Laird, Ed., Kaufmann, pp. 339–352.

[40] Quinlan, J. R. (1987). Decision trees as probabilistic classifiers. In *Proc. 4th International Workshop on Machine Learning* Ca, P. Langley, Ed., Kaufmann.

[41] Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess endgames. In *Machine Learning, vol. 1*, J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, Eds., Tioga, Palo Alto, Ca.

[42] Reisbeck, C. (1981). Failure-driven reminding for incremental learning. In *IJCAI-81* .

[43] Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence 13*, 1–2, 81–132.

[44] Rendell, L. (1988). Learning hard concepts. In *Proc. Third European Working Session on Learning (EWSL-88)* London, D. Sleeman, Ed., Pitman, pp. 177–200.

[45] Rich, E. (1983). *Artificial Intelligence.* McGraw-Hill, NY.

[46] Russell, S. J. (1988). A sketch of automous learning using declarative bias. In *International workshop on Machine Learning, Meta-reasoning and Logics* Portugal, P. Brazdil, Ed., Faculdade de Economia, Univ. Porto, pp. 147–166. (Proceedings to be published in book form in 1989).

[47] Russell, S. J. (1988). Tree-structured bias. In *AAAI-88* Ca, Kaufman, pp. 641–645.

[48] Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence 5*, 2, 115–135.

[49] Sammut, C. (1988). Logic programs as a basis for machine learning. In *International workshop on Machine Learning, Meta-reasoning and Logics* Portugal, P. Brazdil, Ed., Faculdade de Economia, Univ. Porto, pp. 183–193. (Proceedings to be published in book form in 1989).

[50] Sammut, C., and Banerji, R. (1986). Learning concepts by asking questions. In *Machine Learning, vol. 2*, J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, Eds., Tioga, Palo Alto, Ca.

[51] Schank, R., and Abelson, R. (1977). *Scripts, Plans, Goals and Understanding.* Erlbaum, Hillsdale, NJ.

[52] Shapiro, A. D. (1987). *Structured Induction in Expert Systems.* Turing Inst. Press, in association with Addison-Wesley, Wokingham, UK.

[53] Shapiro, E. (1983). *Algorithmic Program Debugging.* MIT Press. (ACM Distinguished Dissertation).

[54] Sussman, G. J. (1975). *A Computer Model of Skill Acquisition.* Elsevier, NY.

[55] VanHarmelen, F., and Bundy, A. (1988). Explanation-based generalisation = partial evaluation. *Artificial Intelligence 36*, 3, 401–412.

[56] VanLehn, K. (1986). Learning a domain theory by completing explanations. In *Machine Learning: A Guide to Current Research*, T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, Eds., Kluwer, Lancaster, UK.

[57] Winston, P. H. (1986). Learning by augmenting rules and accumulating censors. In *Machine Learning, vol. 2*, J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, Eds., Tioga, Palo Alto, Ca, pp. 45–61.